**Topic :->Laravel Task Manager App using Rest API and POSTMAN**

* ***ABSTRACT*** :-

This is a Web Development application where a user can add a task in the application which he/she need’s that task to be done, he/she can mark the task as done when they have completed the work. Now the task may be as Simple as bringing Groceries to Many Important task’s Such as Bringing Medicine. This application is built on some main things that are HTML This is a Markup language that means that this language is used as the GUI which is being interacted by the user. The next main thing is the PHP it is a server-side Scripting Language where this language would take your input from the user it will then used for storing in the database which is the third part where we can store the data needed and delete the data which when done. Then in order for all this to work we have added another main thing which is the laravel Framework.This application provides a smooth and efficient way of manipulation of the task’s and to top off this application is also scalable where it can be used for managing list of larger size. Once you add or mark your task as done you will receive a email saying that the task is created or the task is marked.

* ***OBJECTIVE***:-

The main reason as to why I built this application was to help the user’s stay organized and on top of their tasks. This application can be used mainly for “Increasing productivity” which would help users to stay focused on their tasks and avoid distractions. By Having a clear list of what needs to be done ,users can be better prioritize their time and energy. Then we have another reason which is “Reduced stress and anxiety” now when you have a workload amount of tasks assigned you need to have a diary which can help you keep track as to where you are with your task. Now in olden days people had diary’s to keep the track of task but in this modern world people need more reliable and efficient way of handling the task. We also took in consideration of the users “Time Management” why? Because when you have a list of tasks at your hand the very first thing you do is prioritize as to which task has to be done first and which task is to be taken care after. By having a clear plan for what needs to be done the users can avoid wasting time on unnecessary tasks or activities. Another main reason behind building this project is that the user can increase the sense of accomplishment where this application can help users feel more accomplished by giving them a visual representation of their list, they can see how much they have achieved. The Overall reason behind building this application is to make users more productive reduce the stress and anxiety, improve their time management skills and feel more accomplished. These are the main things that we kept in mind and we actually built the application on that we also included email thing where once the user adds any task or marks any task as done he/she will recive a mail saying to the user that the task is created or the task has been marked and we will also add a date of completion that means we will add a ETA date where the user has to complete a particular task before that date.

* ***INTRODUCTION***:-

Listing application is actually a very simple application whose main motive is to add or assign a list of to do things that need to be done the tasks may be anything like being personal task such as running an errand or cleaning the house ,or for work tasks, such as completing projects or attending meetings, the tasks can be simple or complex ,depending on the individual’s needs and preference .In this era the work load is actually increasing day by day and they need a application where they can add all the tasks now in the olden days people had to write everything down and when something were to happen like their diary in which they had written the tasks were to be gone then they would have no idea as to what they have tasks which can be important. Now a days all the things are getting digitalized where everything is moving online and people are all using the online applications now this website is for people like them who are coming on online. We actually divided this project into 3 parts where the first part would be the “Front-End” or some people also like to say that this is the “GUI ” where the user will interact with the application and will actually add or mark or delete the task according to his wish. The second part is actually the middle ware part where all the things get to work the language which was actually used is “PHP” now the php is actually a server side scripting language now what does that mean that means this actually runs on the server side or you can say the company side will handle all the task after interacting with the user. Then comes the last part “Storage” where all your data is actually stored all the tasks which you are adding and all the tasks that the user is deleting are deleted from the database.

* ***METHODOLOGY :-***

List Application:-

Now when you fire up the application you will see a web page where you will have an heading task manager below that you will have a button called as add a task below that you will have a table in which you have five columns first column is the task number the second one is task description the third one is task owner the 4th one is the estimation time the fifth is the action then when you click on the address button you will have four text box where in the first you will have a description in which you will write the task description of what you will be doing then in the second textbox you will have the name of the owner after that you will have the third textbox which will be email of the owner to whom you are sending the email to then we have an estimate time to complete the particular task after that once when you add a particular task you can have the authority to change that task. In the rest you will have the same attributes just as add a task the only thing that will be added is the task status what is task status? Task status is nothing but the task progress you will have two stages in it the first is the task is in progress and another one is done the task in progress means that you are still doing the task but task status is done you can mark the task as done after that you will get an alert message saying that the task has been edited after that you will have another icon which is called as mark the task as done icon next to Mark as done is delete icon

Database Connection:-

The database it is a storage unit where you can store all the data which your entering while creating a task or editing the task, deleting the task, marking the task as done now the programming language that we use for this database is My SQL in this database we have a table called as tasks in which you will have 8 attributes first one is the ID then task description task owner task owner email task ETA status created at and updated at iD no what does ID contain is the task number it is unique and not null Then we have task description in which the user will be entering the task that he has to do ,task owner in which you will enter the owner of that particular task task owner email contains the email of the task owner then task-ETA this tenses the approximate date by which you have to complete that particular task then status it means that the work is being done or is it completed then created at keeps the record of when you created that particular task updated at this keeps record of when you edit the particular task . Now when ever you Mark the task as done. The status will be changed when you add the task the status will be automatically recorded as zero then when you Mark the task as done it will be changed to one

Mail:-

When you add a task then an email is sent to the particular task owners email saying that this is the task description and you need to complete it by this time and whenever you are marking the task has done another email is sent to the same owner's email saying the task description has been completed

Complete work flow:-

No before running the application you need to do a couple of things one you need to start the Apache and my SQL server at Xampp then you will go to a particular vs quote in which you will open your code then you will open the terminal in this you will create two terminals the one will be where you run the server the other will be where you will be sending the emails using the jobs and queue now you got to type command "php artisan serve" what will happen is the server will start at a particular port location and it will give you an link now you have to tap on that and it will take you to the default Browser's particular url provided by the laravel server now in the other terminal you need to type the command "php atisan queue:work" now what will happen here is it will listen to the job queue when ever you create a new task the particular email address that you have given the work of the job is to send that email and it will listen to that using this command now when you had a task what will happen if you will be sending an email to the particular task owners email saying that the particular task has been created and you need to completed by this time now in that particular row where you want to edit the task you are going to have an icon when you press that I call and alert box will popped up asking you to task description that you want to change the owner of the task that you want to change the email of the owner that you want to change the estimated time to complete that you have to change and the task status from in progress to done or still and progress or none of the above once you have edited you can submit it then you will have another icon which is Mark as done now when you press the button you will get an alert box asking whether are you sure you want to mark this task as done if you click submit then the status will be changed to one and you can observe the particular task that you have marked as done from the task ID till the task eTA everything will be strike off now when you do not have any use of that task you can simply click on the delete icon it will ask you whether are you sure you want to delete this task and when you click on submit button the task will be removed from the database and from the front end

* ***Code:-***

Web.php:

<?php

use Illuminate\Support\Facades\Route;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', function () {

    return ['Laravel' => app()->version()];

});

require \_\_DIR\_\_.'/auth.php';

Route::get('/task',function(){

    return view('tasks.index');

});

Index.blade.php:-

@extends('layout.baseview')

@section('title','All Tasks')

@section('style')

<style>

    .done{

        text-decoration: line-through;

    }

</style>

@endsection

@section('content')

@include('Layout.navigation')

<center><h1>Task Manager</h1></center>

<div class="container mt-5">

    <button type="button" class="btn btn-outline-primary mb-5 end-0" onclick="addtask()">Add A Task</button><br><br>

    <div class="card">

        <div class="card-body">

            <table class="table">

                <thead>

                    <tr>

                        <th scope="col">Si no</th>

                        <th scope="col">Task Description</th>

                        <th scope="col">Task Owner</th>

                        <th scope="col">Task Eta</th>

                        <th scope="col">Action</th>

                    </tr>

                </thead>

                <tbody id="taskTable">

                </tbody>

            </table>

        </div>

    </div>

</div>

<div class="modal fade" id="createTaskModal" tabindex="-1">

    <div class="modal-dialog">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title">Add-Task</h5>

                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

            </div>

            <div class="modal-body">

                <form >

                    <div class="form-group">

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

                        <input type="text" class="form-control" id="createTaskDescription" placeholder="Enter a task"><br>

                    </div>

                    <div class="form-group">

                        <label for="createTaskDescription">Name Of the Owner</label>

                        <input type="text" class="form-control" id="createTaskOwner" placeholder="Enter Task Owner"><br>

                    </div>

                    <div class="form-group">

                        <label for="createTaskDescription">Email of the Owner</label>

                        <input type="text" class="form-control" id="createTaskEmail" placeholder="Enter Task Owner Email"><br>

                    </div>

                    <div class="form-group">

                        <label for="createTaskDescription">Estimate Time to Complete</label>

                        <input type="date" class="form-control" id="createTaskETA" placeholder="Enter Task ETA"><br>

                    </div>

                </form>

            </div>

            <div class="modal-footer">

                <button type="button" class="btb btn-secondary" data-bs-dismiss="modal"> Close</button>

                <button type="button" class="btb btn-primary" onclick="createTask()">Submit</button>

            </div>

        </div>

    </div>

</div>

<div class="modal fade" id="editTaskModal" tabindex="-1">

    <div class="modal-dialog">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title">Edit Task Status</h5>

                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

            </div>

            <div class="modal-body">

                <form >

                    <div class="form-group">

                        <label for="editTaskDescription">Task Description</label>

                        <input type="text" class="form-control" id="editTaskDescription" placeholder="Enter a task">

                    </div>

                    <div class="form-group">

                        <label for="editTaskDescription">Task Owner</label>

                        <input type="text" class="form-control" id="editTaskOwner" placeholder="Enter Task Owner">

                    </div>

                    <div class="form-group">

                        <label for="editTaskDescription">Email of the Owner</label>

                        <input type="text" class="form-control" id="editTaskEmail" placeholder="Enter Task Owner Email">

                    </div>

                    <div class="form-group">

                        <label for="editTaskDescription">Estimated Time to Complete</label>

                        <input type="date" class="form-control" id="editTaskETA" placeholder="Enter Task ETA">

                    </div>

                    <div class="form-group">

                        <label for="editTaskStatus">Task Status</label>

                        <select class="form-control" name="" id="editTaskStatus">

                            <option>Set Task Status</option>

                            <option value="0">In Progress</option>

                            <option value="1">Done</option>

                        </select>

                    </div>

                    <input type="hidden" id="editTaskid">

                </form>

            </div>

            <div class="modal-footer">

                <button type="button" class="btb btn-secondary" data-bs-dismiss="modal"> Close</button>

                <button type="button" class="btb btn-primary" onclick="updateTask()">Submit</button>

            </div>

        </div>

    </div>

</div>

<div class="modal fade" id="doneTaskModal" tabindex="-1">

    <div class="modal-dialog">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title">Mark Task as Done</h5>

                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

            </div>

            <div class="modal-body">

                <p>Are you sure you want to mark this task as Done? </p>

                    <input type="hidden" id="doneTaskid">

            </div>

            <div class="modal-footer">

                <button type="button" class="btb btn-secondary" data-bs-dismiss="modal"> Close</button>

                <button type="button" class="btb btn-primary" onclick="updateTaskasDone()">Submit</button>

            </div>

        </div>

    </div>

</div>

<div class="modal fade" id="deleteTaskModal" tabindex="-1">

    <div class="modal-dialog">

        <div class="modal-content">

            <div class="modal-header">

                <h5 class="modal-title">Delete Task</h5>

                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

            </div>

            <div class="modal-body">

                <p>Are you sure you want to Delete this Task ?</p>

                    <input type="hidden" id="deleteTaskid">

            </div>

            <div class="modal-footer">

                <button type="button" class="btb btn-secondary" data-bs-dismiss="modal"> Close</button>

                <button type="button" class="btb btn-primary" onclick="updateTaskDelete()">Submit</button>

            </div>

        </div>

    </div>

</div>

@endsection

@section('customjs')

<script>

$(document).ready(function(){

    // console.log(123);

    getAllTasks();

});

function getAllTasks(){

    $.ajax({

        type:'get',

        url:'http://localhost:8000/api/task',

        success:function(result){

            var html='';

            for(var i=0;i<result.length;i++){

                var linethrough=result[i]['status']==1?'class="done"':"";

                html+=  '<tr>'

                        +'<th scope="row" '+linethrough+'>'+(i+1)+'</th>'

                        +'<td '+linethrough+'>'+result[i]['task\_description']+'</td>'

                        +'<td '+linethrough+'>'+result[i]['task\_owner']+'</td>'

                        +'<td '+linethrough+'>'+result[i]['task\_eta']+'</td>'

                        +'<td>'

                            +'<i class="bi bi-pencil-square" onclick="editTask('+result[i]['id']+')"></i>'

                            +'<i class="bi bi-check2-square"  onclick="markasdone('+result[i]['id']+')"></i>'

                            +'<i class="bi bi-trash"  onclick="deletetask('+result[i]['id']+')"></i>'

                            +'</td>'

                            +'</tr>'

                }

            $("#taskTable").html(html)

        },

        error:function(e){

            console.log(e.responseText);

        }

    });

}

function addtask(){

    $('#createTaskModal').modal('show');

}

function createTask(){

    var task\_description=$("#createTaskDescription").val()

    var task\_owner=$("#createTaskOwner").val()

    var task\_owner\_email=$("#createTaskEmail").val()

    var task\_eta=$("#createTaskETA").val()

    $.ajax({

        type:'post',

        url:'http://localhost:8000/api/task',

        data:{

            task\_description:task\_description,

            task\_owner:task\_owner,

            task\_owner\_email:task\_owner\_email,

            task\_eta:task\_eta

        },

        success:function(result) {

            $("#createTaskModal").modal('hide');

            getAllTasks();

        },

        error:function(e){

            console.log(e.responseText);

        }

    })

}

function editTask(id){

    $.ajax({

        type:'get',

        url:'http://localhost:8000/api/task/'+id,

        success:function(result) {

            $("#editTaskDescription").val(result['task\_description']);

            $("#editTaskOwner").val(result['task\_owner']);

            $("#editTaskEmail").val(result['task\_owner\_email']);

            $("#editTaskETA").val(result['task\_eta']);

            $("#editTaskStatus").val(result['status']);

            $("#editTaskid").val(result['id']);

            $('#editTaskModal').modal('show');

                getAllTasks();

            },

            error:function(e){

                console.log(e.responseText);

            },

    })

}

function updateTask(){

    var id=$("#editTaskid").val()

    var task\_description=$("#editTaskDescription").val()

    var task\_owner=$("#editTaskOwner").val()

    var task\_owner\_email=$("#editTaskEmail").val()

    var task\_eta=$("#editTaskETA").val()

    var task\_status=$("#editTaskStatus").val()

    $.ajax({

        type:'put',

        url:'http://localhost:8000/api/task/'+id,

        data:{

            task\_description:task\_description,

            task\_owner:task\_owner,

            task\_owner\_email:task\_owner\_email,

            task\_eta:task\_eta,

            task\_status:task\_status

        },

        success:function(result) {

            $("#editTaskModal").modal('hide');

            getAllTasks();

        },

        error:function(e){

            console.log(e.responseText);

        }

    })

}

function markasdone(id){

    $("#doneTaskid").val(id)

    $("#doneTaskModal").modal('show')

}

function updateTaskasDone(){

    var id=$("#doneTaskid").val()

    $.ajax({

        type:'post',

        url:'http://localhost:8000/api/task/done/'+id,

        success:function(result) {

            $("#doneTaskModal").modal('hide');

            getAllTasks();

        },

        error:function(e){

            console.log(e.responseText);

        }

    })

}

function deletetask(id){

    $("#deleteTaskid").val(id)

    $("#deleteTaskModal").modal('show')

}

function updateTaskDelete(){

    var id=$("#deleteTaskid").val();

    $.ajax({

        type:'delete',

        url:'http://localhost:8000/api/task/'+id,

        success:function(result) {

            $("#deleteTaskModal").modal('hide');

            getAllTasks();

        },

        error:function(e){

            console.log(e.responseText);

        }

    })

}

</script>

@endsection

Navigation.blade.php:

<header class="navbar navbar-dark sticky-top bg-dark flex-md-nowrap p-0 shadow">

    <a href="#" class="navbar-brand col-md-3 col-lg-2 me-0 px-3 fs-6">

        Task Manager

    </a>

    <button class="navbar-toggler position-absolute d-md-none collapsed" type="button" data-bs-toggle="collapse" data-bs-targe="#sidebarmenu">

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

    </button>

    <div class="navbar-nav">

        <div class="nav-item text-nowrap">

            <a href="#">All Tasks</a>

        </div>

    </div>

</header>

Js.blade.php:

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://code.jquery.com/jquery-3.7.1.js" integrity="sha256-eKhayi8LEQwp4NKxN+CfCh+3qOVUtJn3QNZ0TciWLP4=" crossorigin="anonymous"></script>

Css.blade.php:

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.css" integrity="sha512-5A8nwdMOWrSz20fDsjczgUidUBR8liPYU+WymTZP1lmY9G6Oc7HlZv156XqnsgNUzTyMefFTcsFH/tnJE/+xBg==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-icons.css" />

Baseview.blade.php:

<!DOCTYPE html>

<html lang="{{str\_replace('\_','-',app()->getLocale())}}">

    <head>

        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width,initial-scale=1">

        <title>Task Manager | @yield('title')</title>

        @include('layout.css')

        @yield('style')

    </head>

    <body>

        @yield('content')

        @include('layout.js')

        @yield('customjs')

    </body>

</html>

Api.php:

<?php

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\API\ApiTaskController;

/\*

|--------------------------------------------------------------------------

| API Routes

|--------------------------------------------------------------------------

|

| Here is where you can register API routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| is assigned the "api" middleware group. Enjoy building your API!

|

\*/

Route::middleware(['auth:sanctum'])->get('/user', function (Request $request) {

    return $request->user();

});

Route::post('task',[ApiTaskController::class,'create']);

Route::get('task',[ApiTaskController::class,'index']);

Route::get('task/{id}',[ApiTaskController::class,'gettaskbyid']);

Route::put('task/{id}',[ApiTaskController::class,'update']);

Route::post('task/done/{id}',[ApiTaskController::class,'markasdone']);

Route::delete('task/{id}',[ApiTaskController::class,'delete']);

ApiTaskController.php:

<?php

namespace App\Http\Controllers\API;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\TaskManager;

use App\Jobs\SendEmailJob;

class ApiTaskController extends Controller

{

    public function create(Request $request){

        // dd($request->all());S

        $data=new TaskManager;

        $data->task\_description =$request->get('task\_description');

        $data->task\_owner =$request->get('task\_owner');

        $data->task\_owner\_email =$request->get('task\_owner\_email');

        $data->task\_eta =$request->get('task\_eta');

        if($data->save()){

            dispatch(new SendEmailJob($data));

            return "Task saved successfully";

        }

        else{

            return "Something went wrong";

        }

    }

    public function index(){

        $data=TaskManager::get();

        return $data;

    }

    public function gettaskbyid($id){

        $data=TaskManager::find($id);

        return $data;

    }

    public function update(Request $request,$id){

        $data=TaskManager::find($id);

        $data->task\_description =$request->get('task\_description');

        $data->task\_owner =$request->get('task\_owner');

        $data->task\_owner\_email =$request->get('task\_owner\_email');

        $data->task\_eta =$request->get('task\_eta');

        if($data->save()){

            return "Task saved successfully";

        }

        else{

            return "Something went wrong";

        }

    }

    public function markasdone($id){

        $data=TaskManager::find($id);

        $data->status=1;

        if($data->save()){

            dispatch(new SendEmailJob($data));

            return "Task is saved Successfully";

        }

        else{

            return "Task is not saved made mistake some where";

        }

    }

    public function delete($id){

        $data=TaskManager::find($id);

        if($data->delete()){

            return "Task is Deleted Successfully";

        }

        else{

            return "Task is not saved made mistake some where";

        }

    }

}

TaskMail.php

<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Mail\Mailable;

use Illuminate\Mail\Mailables\Content;

use Illuminate\Mail\Mailables\Envelope;

use Illuminate\Queue\SerializesModels;

class TaskMail extends Mailable

{

    use Queueable, SerializesModels;

    private $data;

    /\*\*

     \* Create a new message instance.

     \*/

    public function \_\_construct($data)

    {

        $this->data=$data;

    }

    /\*\*

     \* Get the message envelope.

     \*/

    public function envelope(): Envelope

    {

        return new Envelope(

            subject: 'Task Creation Mail',

        );

    }

    /\*\*

     \* Get the message content definition.

     \*/

    public function content(): Content

    {

        return new Content(

            view: 'email.notification',

            with : ['data'=>$this->data]

        );

    }

    /\*\*

     \* Get the attachments for the message.

     \*

     \* @return array<int, \Illuminate\Mail\Mailables\Attachment>

     \*/

    public function attachments(): array

    {

        return [];

    }

}

SendEmailJob.php

<?php

namespace App\Jobs;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldBeUnique;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Foundation\Bus\Dispatchable;

use Illuminate\Queue\InteractsWithQueue;

use Illuminate\Queue\SerializesModels;

use App\Mail\TaskMail;

use Illuminate\Support\Facades\Mail;

class SendEmailJob implements ShouldQueue

{

    use Dispatchable, InteractsWithQueue, Queueable, SerializesModels;

    private $data;

    /\*\*

     \* Create a new job instance.

     \*/

    public function \_\_construct($data)

    {

        $this->data=$data;

    }

    /\*\*

     \* Execute the job.

     \*/

    public function handle(): void

    {

        $mail=new TaskMail($this->data);

        Mail::to($this->data['task\_owner\_email'])->send($mail);

    }

}

Notification.blade.php

<!DOCTYPE html>

<html>

    <head>

        <title>Mail For You</title>

    </head>

    <body>

        <p>

            Dear {{$data->task\_owner}},<br>

            The task {{$data->task\_description}}

            @if($data->status == 0)

                Has been added for you.<br>

                Kindly Complete it within {{$data->task\_eta}},<br>

            @endif

            @if($data->status == 1)

                Has Been Completed,<br>

            @endif

            Thank You,<br>

            Regards.<br>

        </p>

    </body>

</html>

TaskManager.php

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class TaskManager extends Model

{

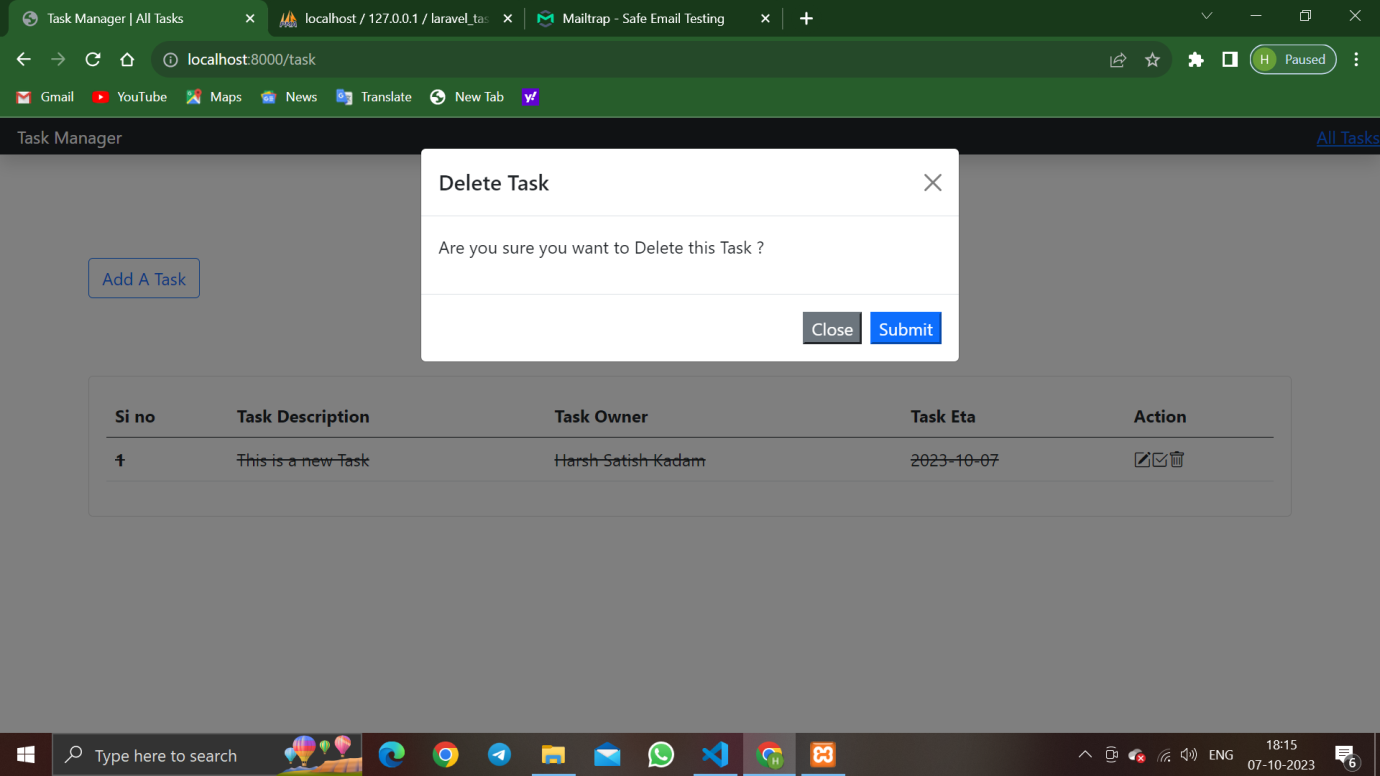
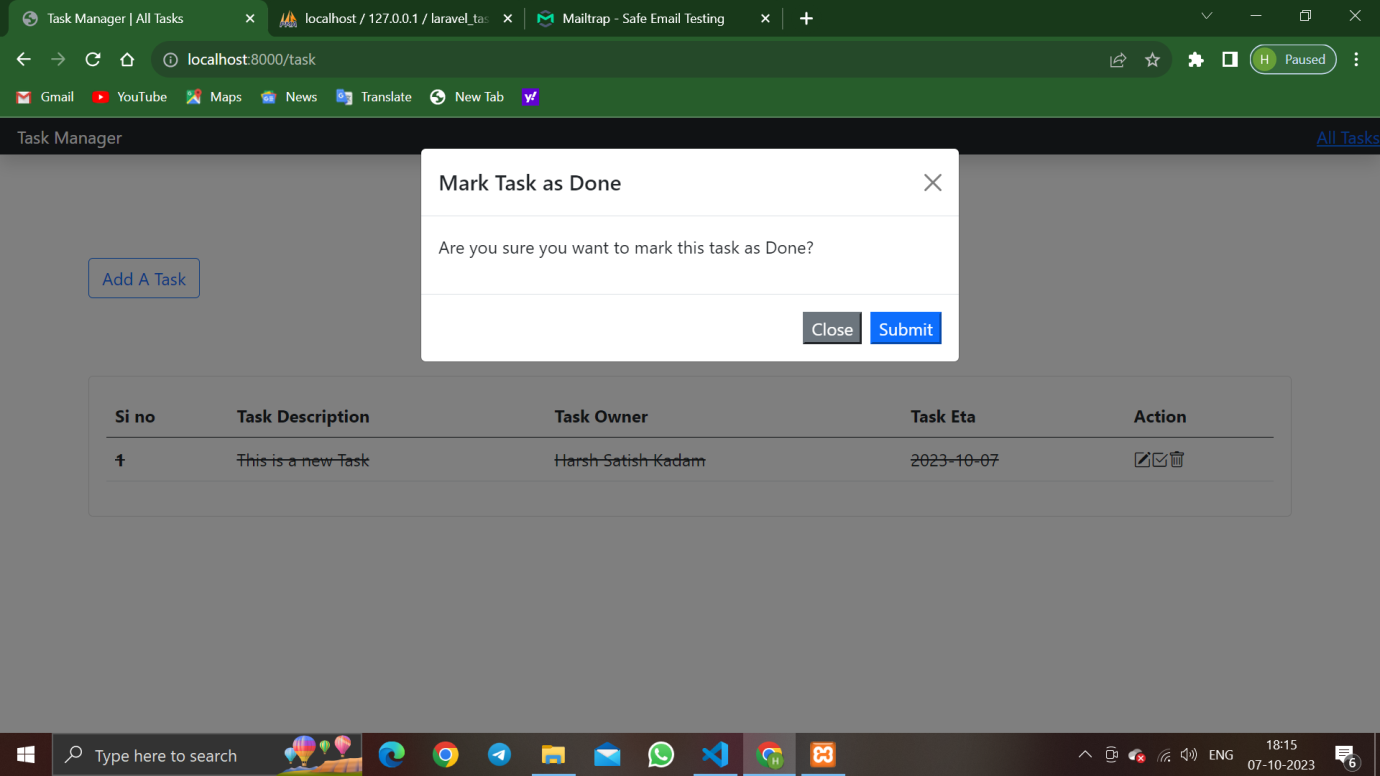
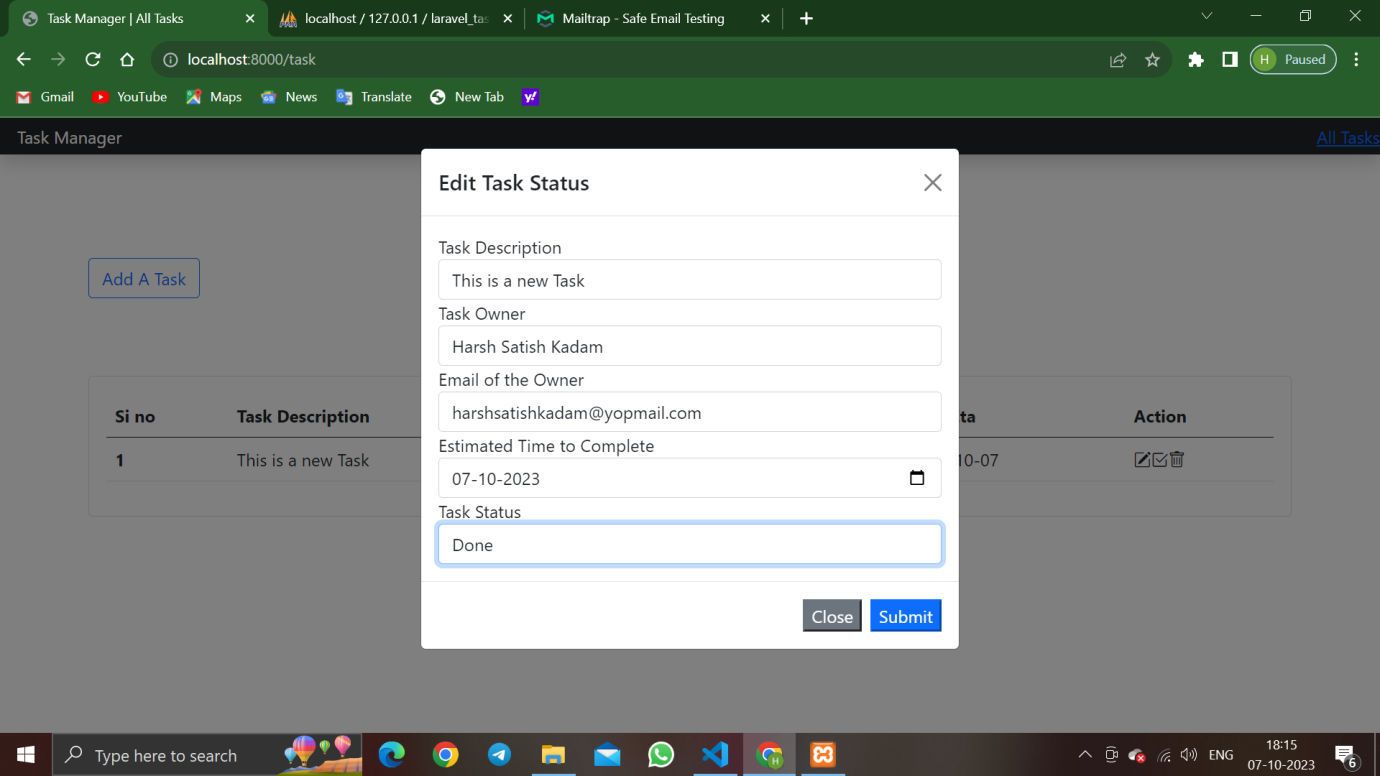
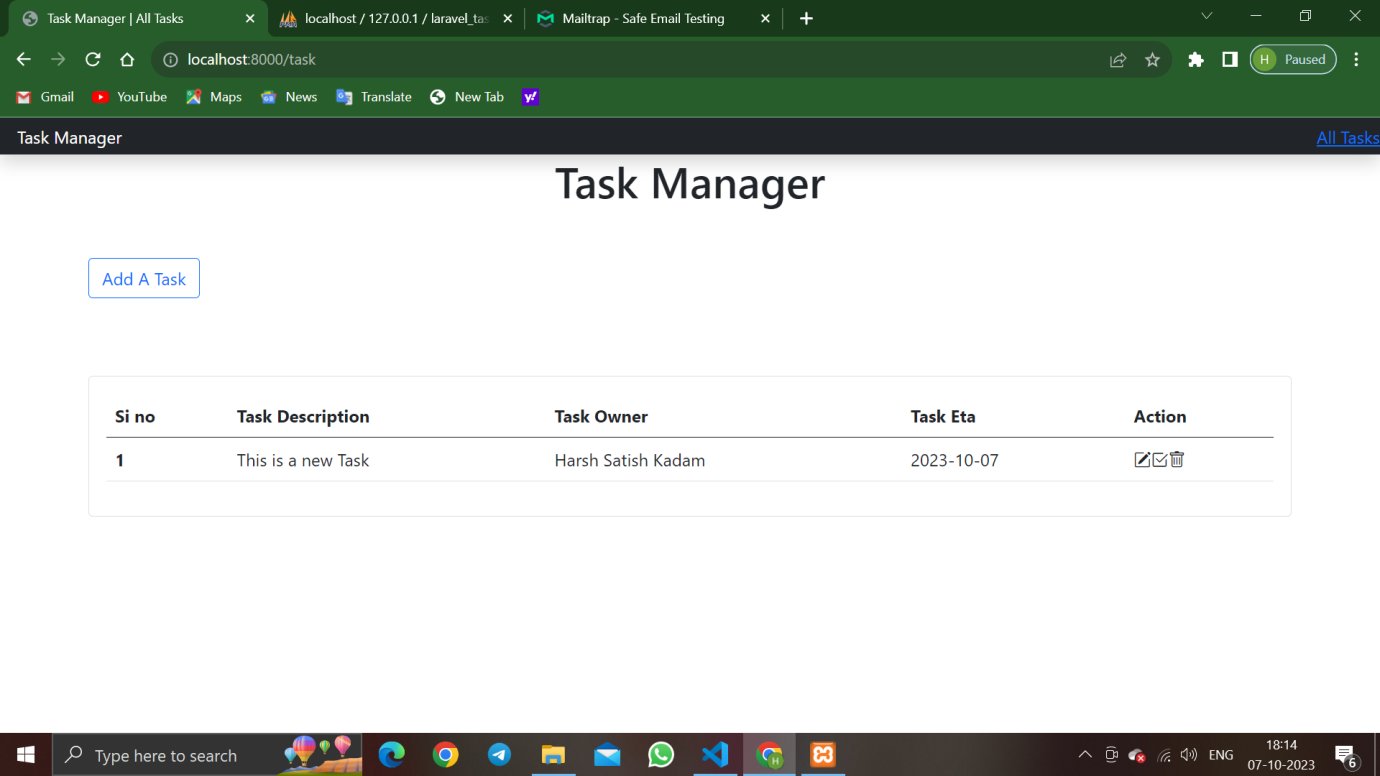
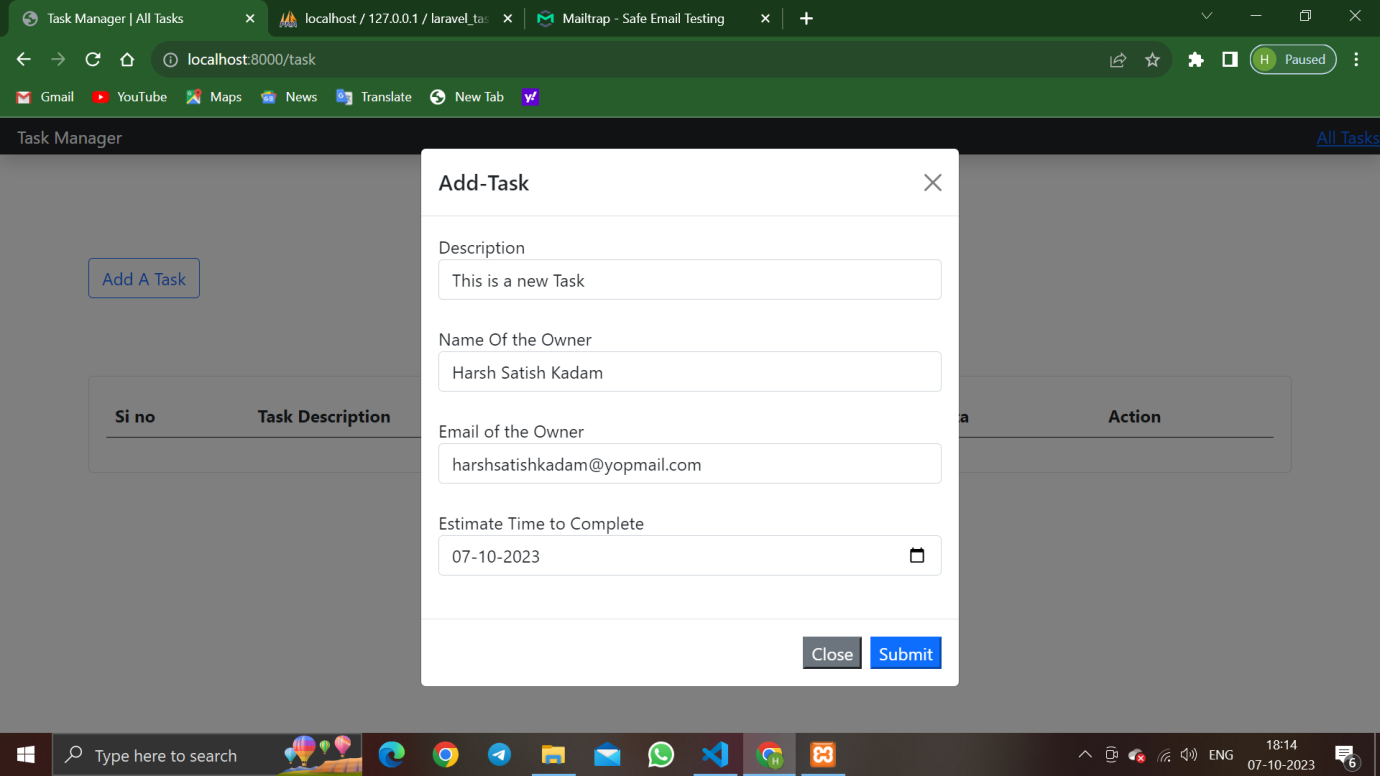
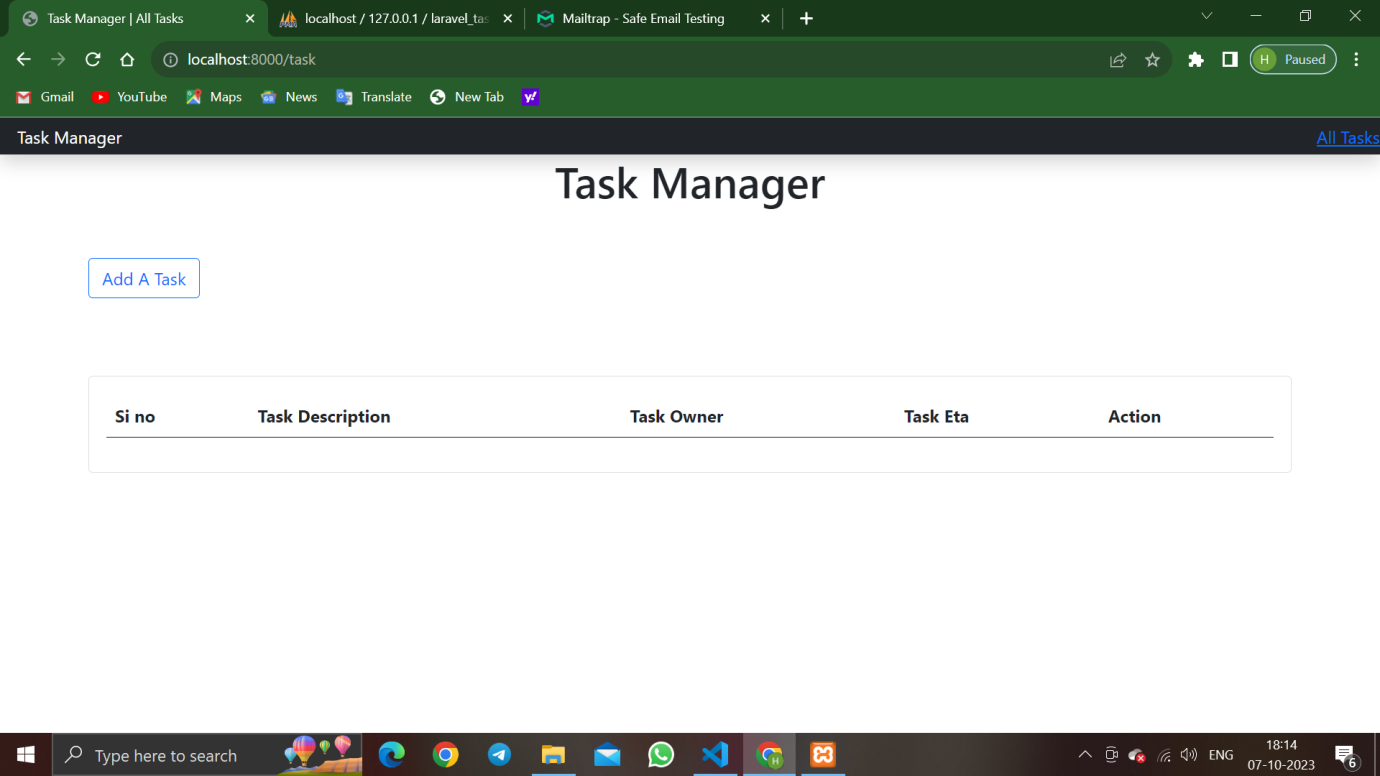
    use HasFactory;

    protected $table='tasks';

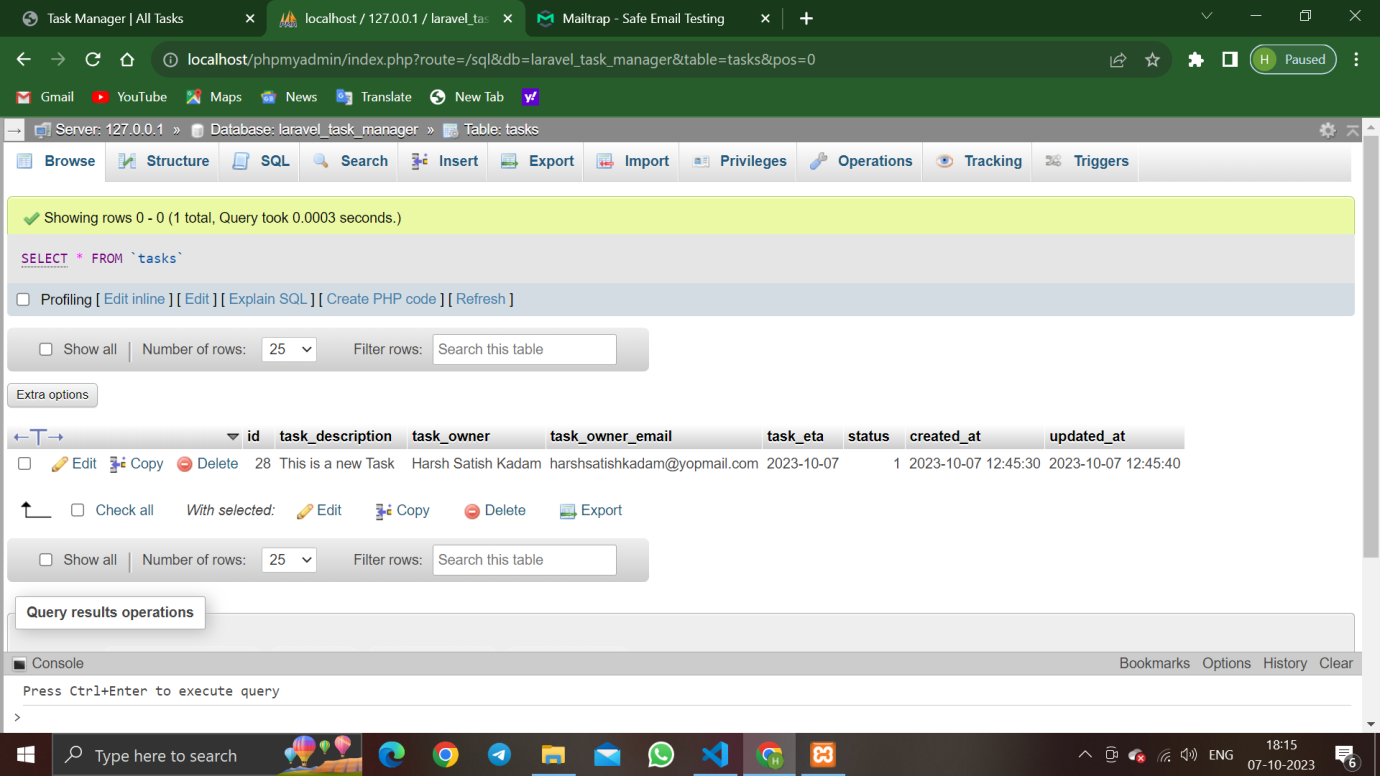
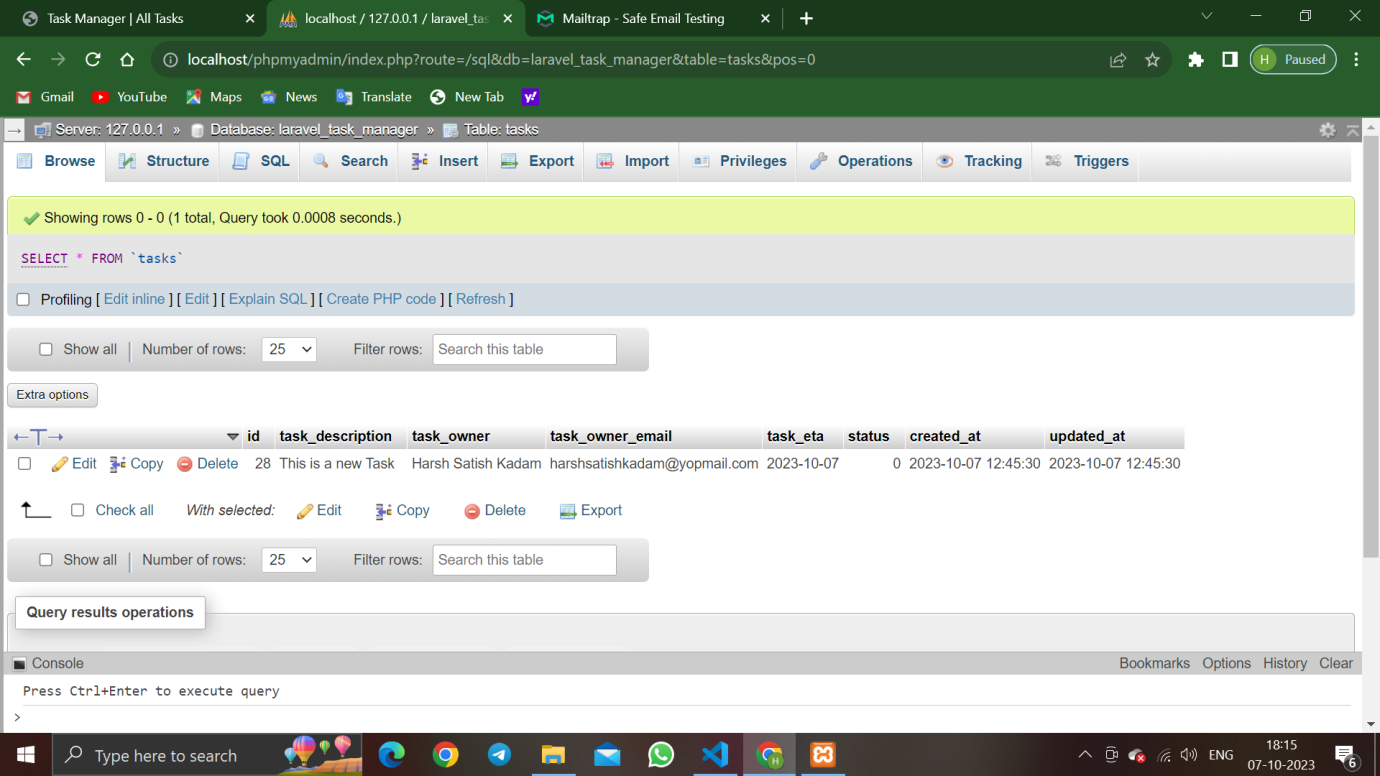
}

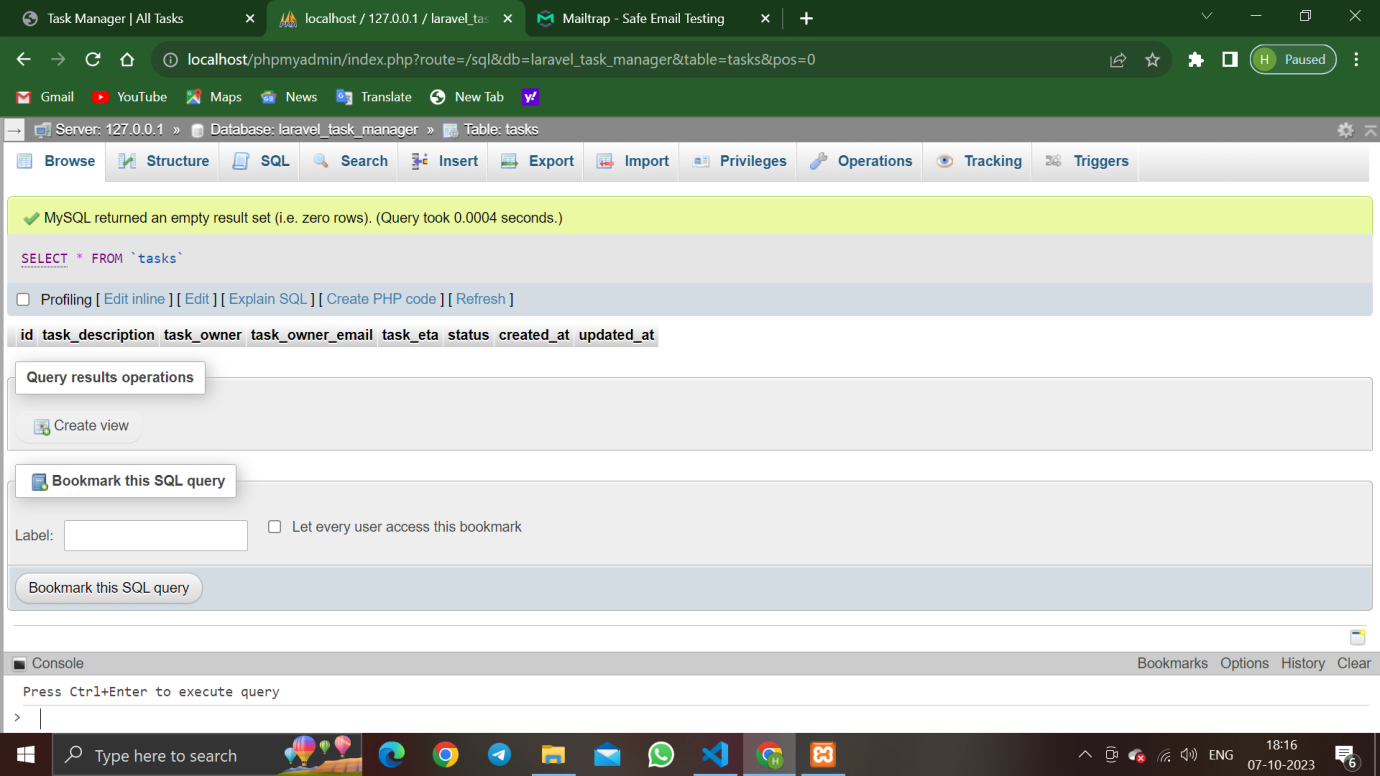
***Outputs:-***

This is the place where you will add,delete,mark the task as done:

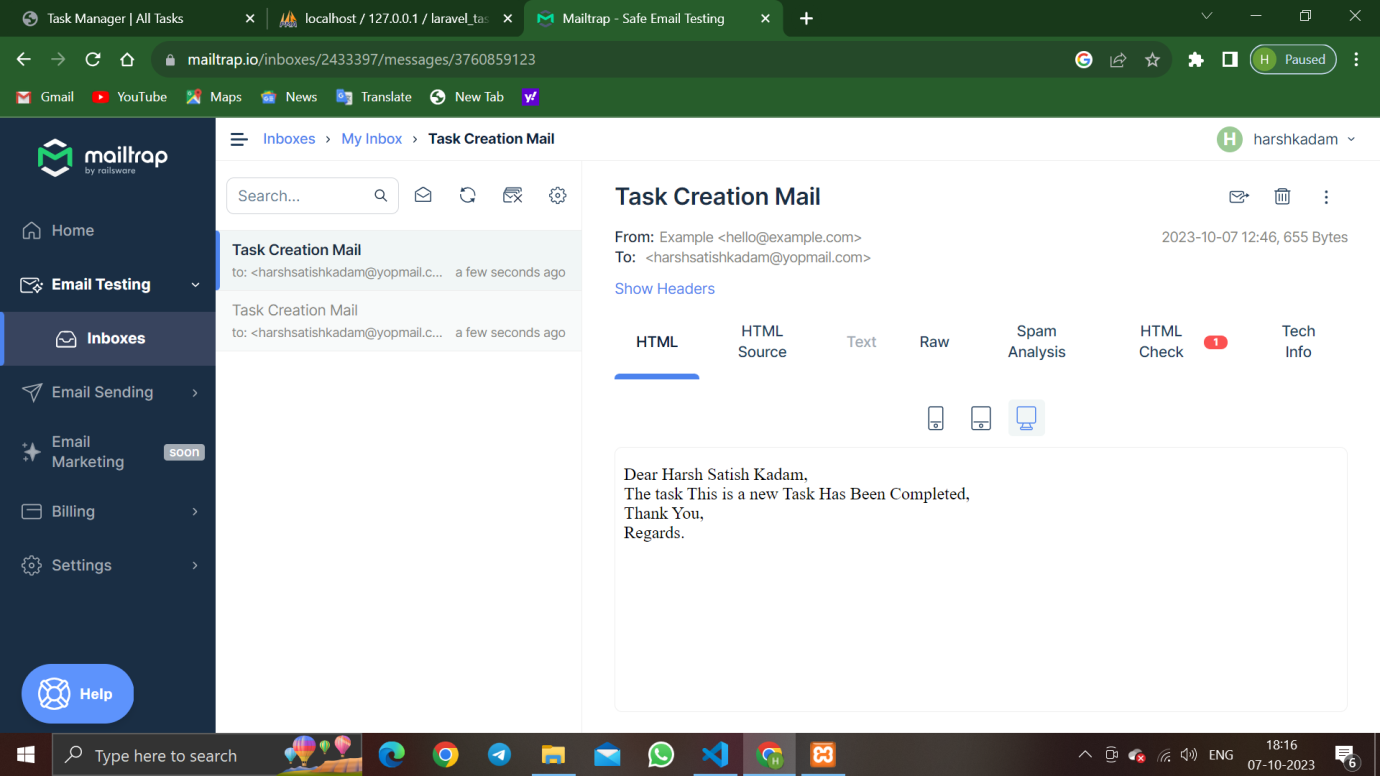
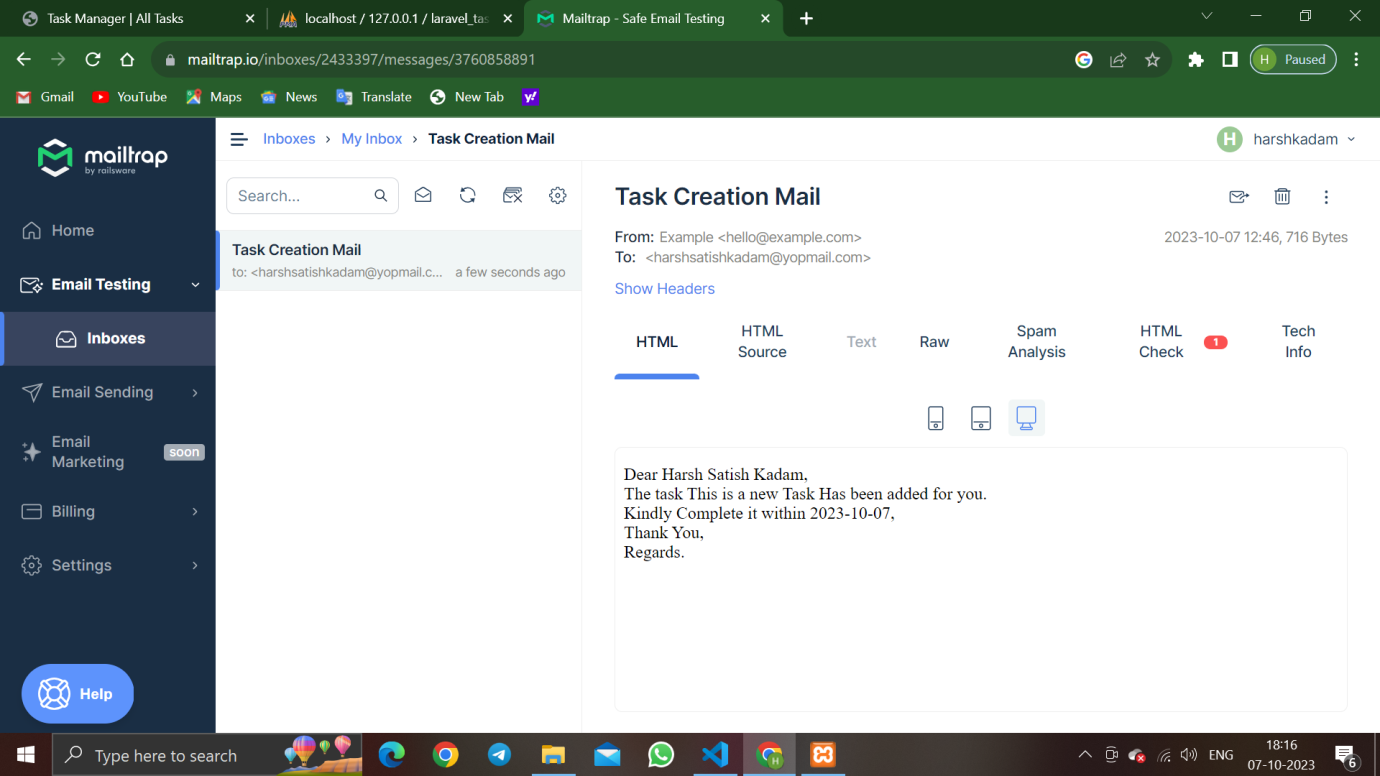


This is the database Connection:





This is the mail application:



***Conclusion:-***

In Co