

Task 1: Project Introduction, Setup and Error Resolution Report

Introduction:

This project is a Laravel framework project for task management. It is created with Laravel php 8, Laravel UI dependency, and MySQL database. The main functionality of this project is Create, Delete, Edit, and view tasks.

Process Flow:

1. Login/ Register to the system
2. Home page (a list of tasks if there is any)
3. Add task => create task page
4. Edit task => edit task page
5. Delete/ Mark task as done => same page
6. Logout

Issue 1: Redirect After Registration to Non-Existent Home Page

Symptom:

After registering, the application redirects to a /home page, resulting in a "Page Not Found" error.

Cause:

Laravel's default behavior is to redirect users to the /home route after registration.

Fix:

Update the `redirectTo` property in the `RegisterController` to redirect to the tasks index route.

```
// File: app/Http/Controllers/Auth/RegisterController.php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
```

```

use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{
    use RegistersUsers;

    protected $redirectTo = '/tasks';

    // rest of the class remains unchanged...
}

```

Issue 2: foreach() argument must be of type array|object, null given

Symptom:

When accessing the tasks page, an error occurs indicating that a `null` value is being passed to `foreach`.

Cause:

The `$tasks` variable is `null` when the user has no tasks.

Fix:

Ensure that `$tasks` is always an iterable collection, even if it's empty.

```

// File: app/Http/Controllers/TaskController.php

namespace App\Http\Controllers;

use App\Models\Task;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class TaskController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function index()
    {
        $tasks = Auth::user()->tasks ?? collect();
        return view('tasks.index', compact('tasks'));
    }

    // Other methods remain unchanged...
}

```

```
}
```

Issue 3: Call to undefined method `App\Models\User::tasks()`

Symptom:

When trying to add a task, an error occurs indicating that the `tasks` method is undefined.

Cause:

The `tasks` relationship is not defined in the `User` model.

Fix:

Define the `tasks` relationship in the `User` model.

```
// File: app/Models/User.php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    public function tasks()
    {
        return $this->hasMany(Task::class);
    }
}
```

Additionally, ensure the inverse relationship is defined in the `Task` model:

```
// File: app/Models/Task.php

namespace App\Models;

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

class Task extends Model
{
    use HasFactory;
}
```

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

Ensure the `tasks` table has a `user_id` column:

1. Create a migration:

```
php artisan make:migration add_user_id_to_tasks_table --table=tasks
```

2. Update the migration file:

```
public function up()
{
    Schema::table('tasks', function (Blueprint $table) {
        $table->foreignId('user_id')->constrained()->onDelete('cascade');
    });
}

public function down()
{
    Schema::table('tasks', function (Blueprint $table) {
        $table->dropForeign(['user_id']);
        $table->dropColumn('user_id');
    });
}
```

3. Run the migration:

```
php artisan migrate
```

Update the `store` method in `TaskController` to associate the task with the authenticated user:

```
// File: app/Http/Controllers/TaskController.php

namespace App\Http\Controllers;

use App\Models\Task;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class TaskController extends Controller
{
    public function store(Request $request)
```

```

{
    $request->validate([
        'title' => 'required|string|max:255',
        // other validation rules...
    ]);

    $task = new Task([
        'title' => $request->input('title'),
        // other fields...
    ]);

    Auth::user()->tasks()->save($task);

    return redirect()->route('tasks.index')->with('success', 'Task created successfully.');
```

Instructions to Open the Project

Prerequisites

- PHP (>= 7.3)
- Composer
- Node.js and npm
- A web server (e.g., Apache, Nginx)
- Database (e.g., MySQL, SQLite)

Steps

1. Download the Project

- If the project is from GitHub:

```
git clone
https://github.com/ahmedalafoori/task_management_CRUD_Laravel.git
```

- If the project is from an email, download the zip file, extract it, and navigate into the project directory:

```
cd path-to-extracted-folder
```

2. Install Dependencies

- Install PHP dependencies using Composer:

```
composer install
```

- Install JavaScript dependencies using npm:

```
npm install
```

3. Environment Configuration

- Copy the .env.example file to .env:

```
cp .env.example .env
```

- Generate the application key:

```
php artisan key:generate
```

- Configure the .env file with your database and other environment settings.

4. Database Setup

- Run the migrations to set up the database:

```
php artisan migrate
```

5. Serve the Application

- Run the built-in PHP server:

```
php artisan serve
```

- Alternatively, configure your web server to serve the Laravel project.

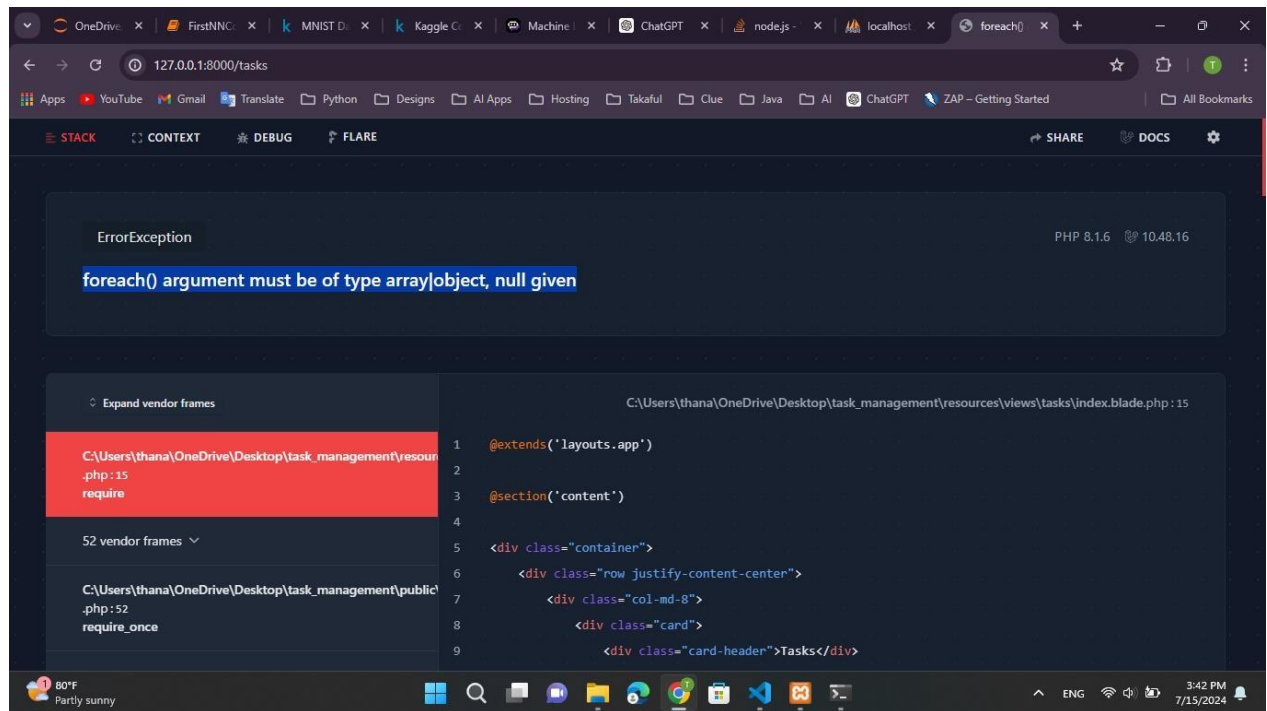
6. Access the Application

- Open your web browser and navigate to <http://localhost:8000> (or the configured web server URL).

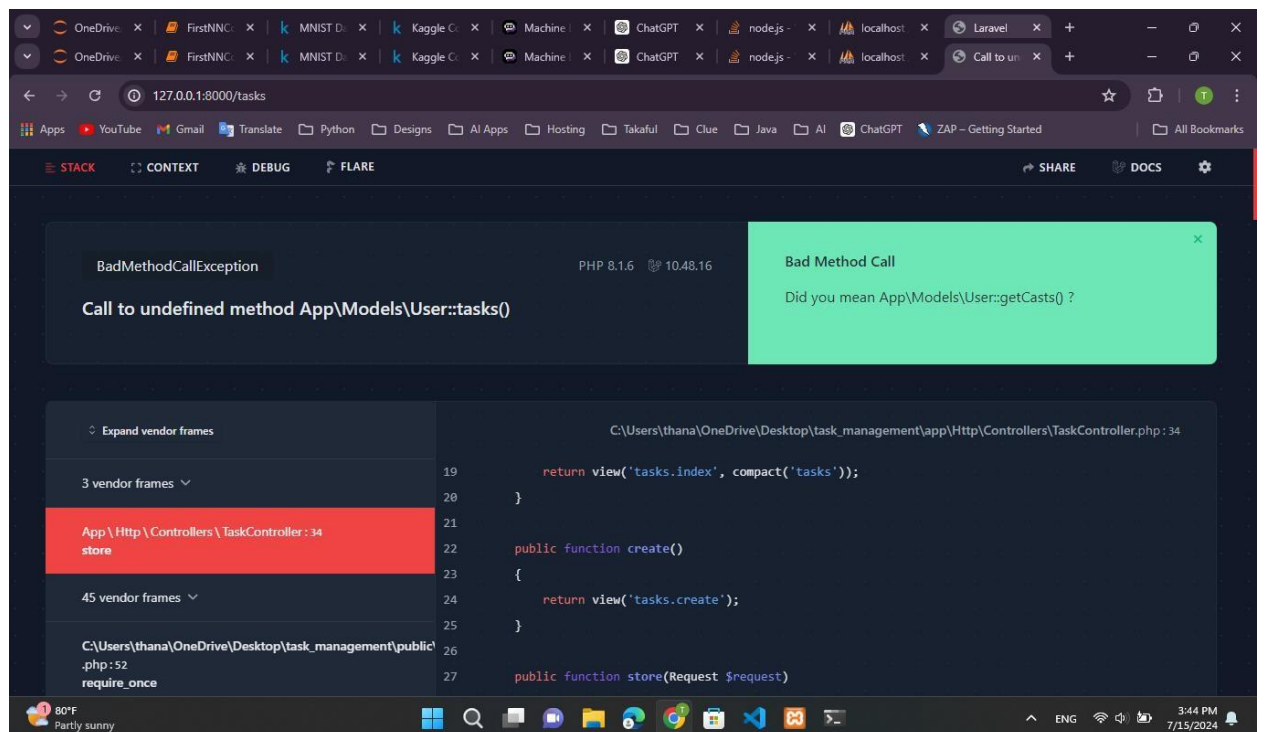
Project GitHub Link:

https://github.com/ahmedalafoori/task_management_CRUD_Laravel.git

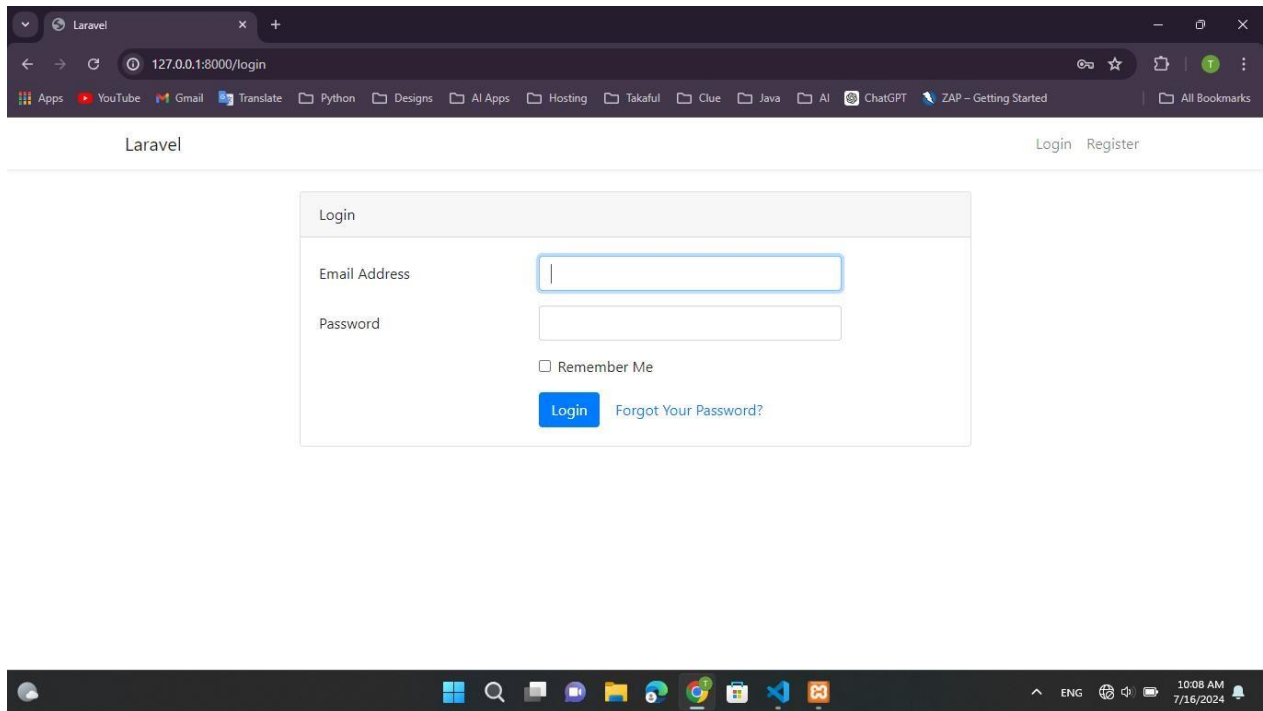
Project Screenshots and UI



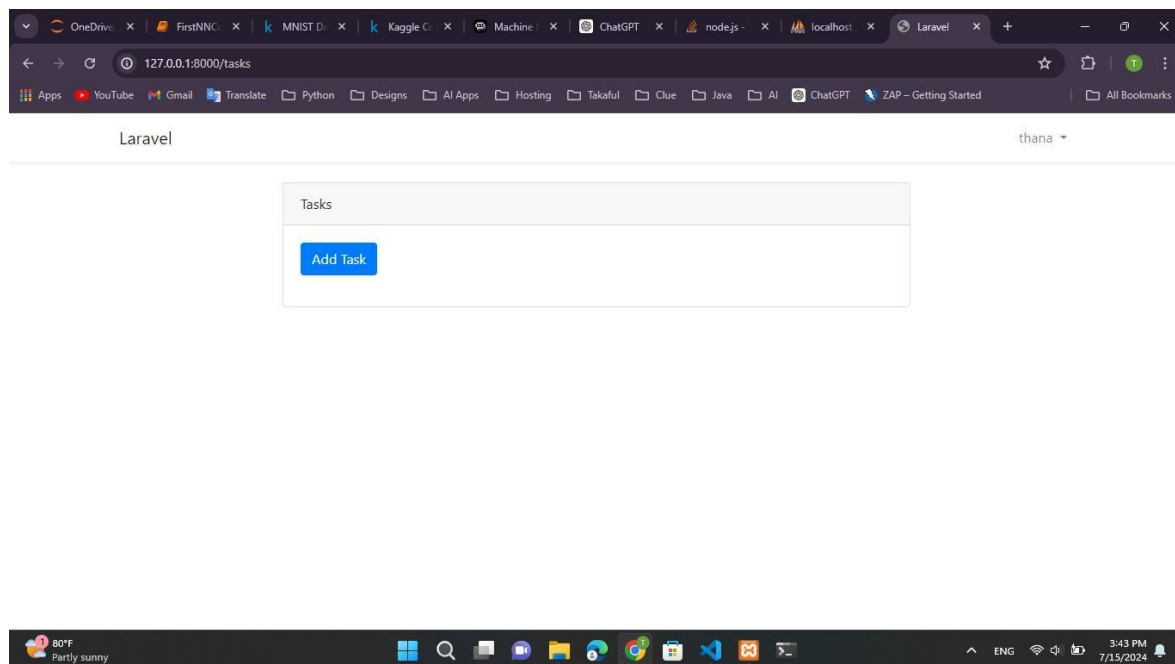
Issue No.2



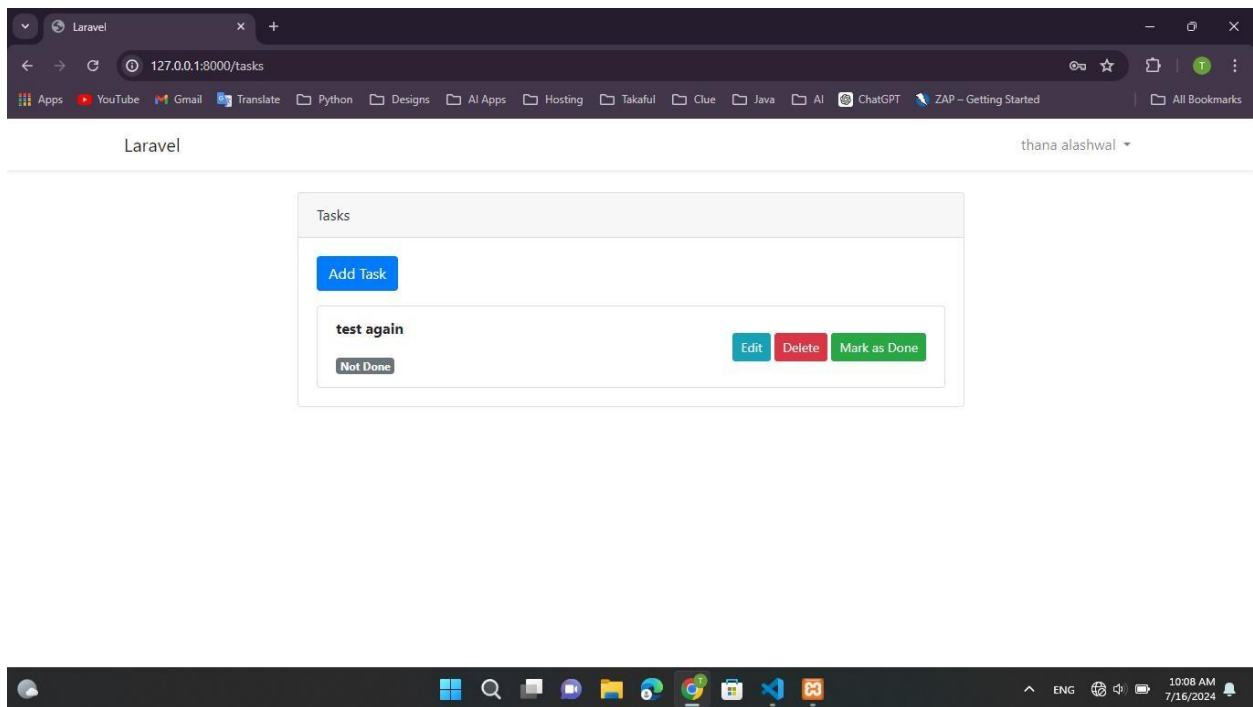
Issue No.3



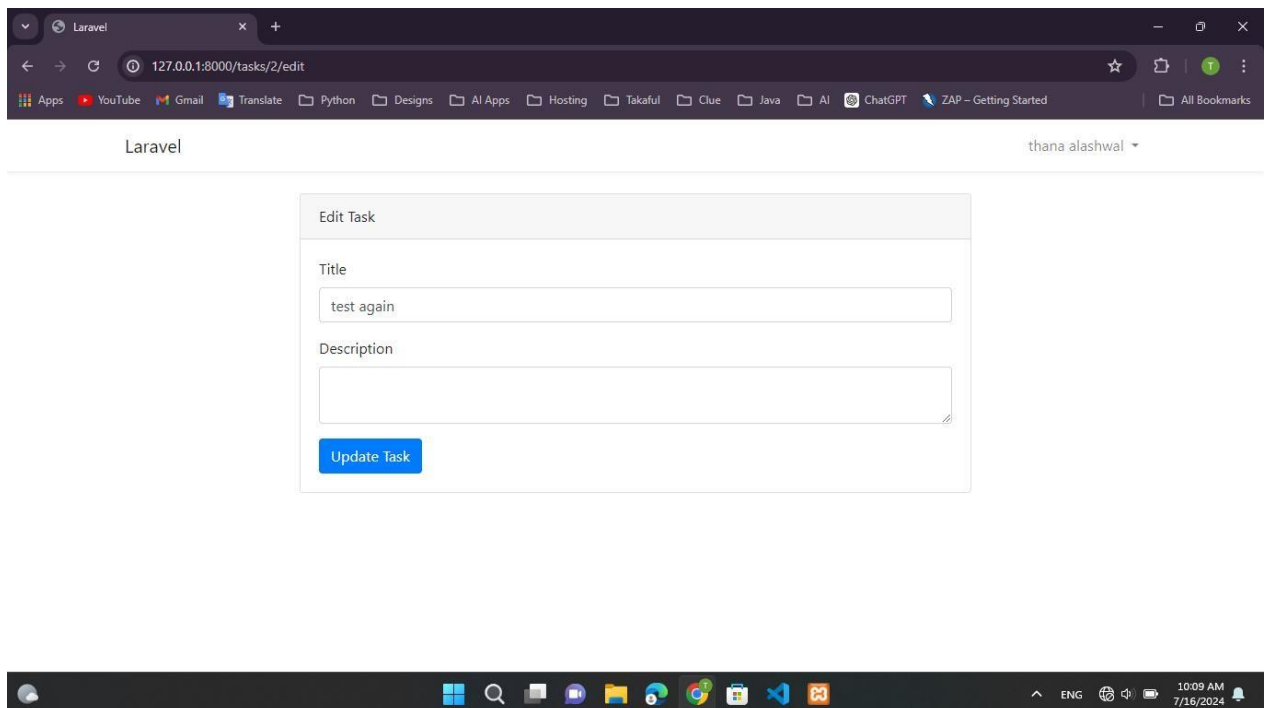
Login/Register Page



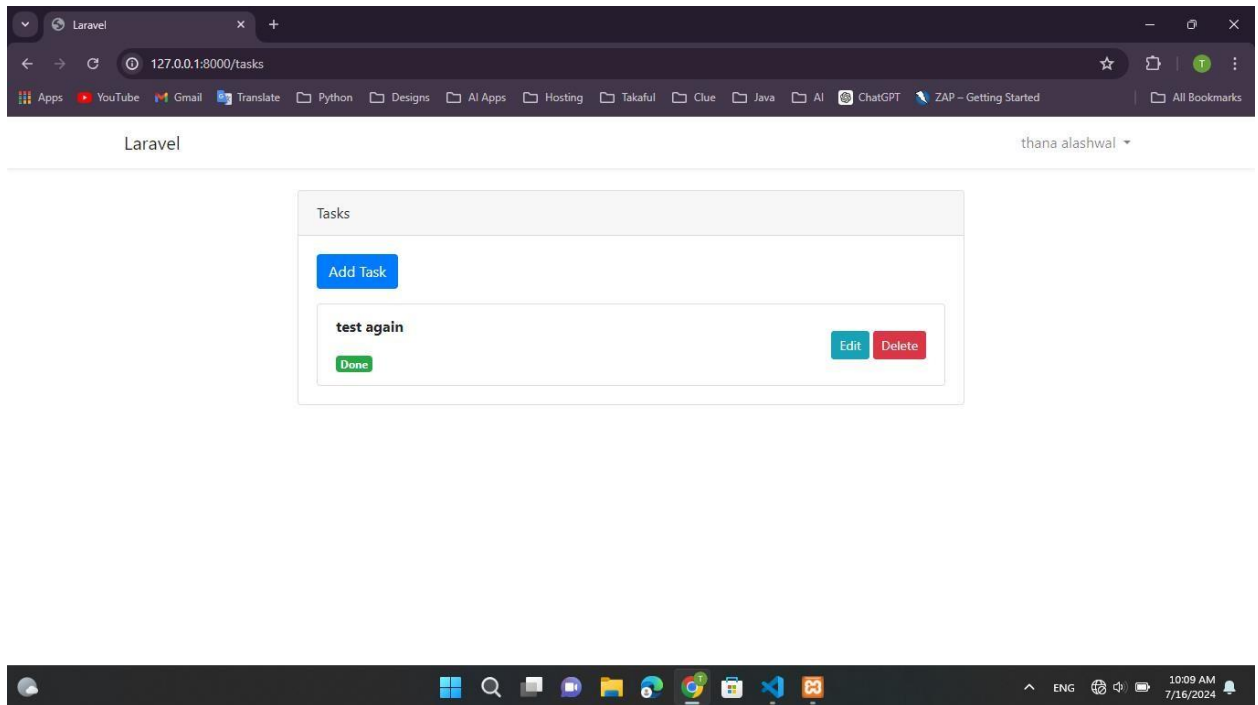
Tasks Page



Tasks Page/Tasks List



Edit Tasks Page



Task Marked as Done