UNIVERSITY INSTITUTE OF COMPUTING

Project on

“COMPLAINT MANAGEMENT SYSTEM”

Program Name: BCA

SUBJECT NAME: ADVANCED WEB DEVELOPMENT

SUBJECT CODE: 23CAH-302

Submitted To: Mr.Vinod Bagga

Designation: Assistant Professor

Submitted By : Anjali

Uid : 23bca10626

Section : 3

Group : A

Abstract

The Complaint Management System is a web-based application designed to streamline the process of lodging, tracking, and resolving complaints efficiently. It allows users to register complaints online and monitor their status in real time. The system provides administrators with tools to manage complaints, assign tasks, and update resolution progress. It ensures transparency and accountability between users and the administration. Built using **Laravel, PHP, MySQL, CSS, and JavaScript**, the system offers a user-friendly interface and secure data handling. The dashboard displays summaries of total, pending, and resolved complaints. Notifications keep users informed about updates. The system reduces manual work and response time. It enhances communication between users and administrators.

Additionally, the project incorporates role-based access control for users and admins. It provides a centralized platform for record management and complaint analysis. The system supports efficient data retrieval and report generation for better decision-making. It contributes to digital transformation by replacing paper-based complaint handling. Overall, it ensures a reliable, transparent, and efficient complaint resolution process.

The project emphasizes usability, scalability, and security in complaint handling. It can be customized for institutions, organizations, or public service sectors. The use of modern web technologies ensures responsive design and accessibility. Regular database backups ensure data integrity and safety. In conclusion, the Complaint Management System represents an innovative approach to managing complaints effectively and systematically.

Introduction

In many organizations, manual complaint handling can be slow, inefficient, and prone to errors. The **Complaint Management System** provides a digital solution for managing complaints effectively. Users can submit complaints with details, and administrators can respond promptly, ensuring all issues are documented and resolved. This system promotes accountability, reduces delays, and allows for real-time tracking of complaints.

Objectives

- Simplify complaint submission for users.
- Enable administrators to manage and respond to complaints efficiently.
- Maintain records of complaints for accountability and reporting.
- Provide notifications and status updates to users.

Problem Statement/Objectives

In many organizations, complaints are handled manually, which often leads to delays, miscommunication, lost records, and lack of accountability. Users may not know the status of their complaints, and administrators may struggle to track and resolve issues efficiently. There is a need for an automated system that allows users to submit complaints, track their status, and enables administrators to manage and resolve them effectively.

Objectives

1. Develop a **user-friendly interface** for submitting complaints.
2. Enable users to **track the status** of their complaints in real-time.
3. Allow administrators to **manage, respond to, and resolve complaints** efficiently.
4. Maintain a **record of all complaints and responses** for accountability and reporting.
5. Improve **communication, transparency, and efficiency** in the complaint handling process.

5. System Design / Approach

The **System Design / Approach** section explains how your Complaint Management System is **organized and implemented**. It includes the architecture, flow of data, design diagrams, and logic.

1. System Architecture

The system follows a **three-tier architecture**:

1. **Presentation Layer (Frontend)** ○ Built using **HTML, CSS, and JavaScript**.
   ○ Provides user-friendly interfaces for users and admins to interact with the system.
2. **Business Logic Layer (Backend)** ○ Developed using **Laravel (PHP framework)**. ○ Handles the processing of user requests, complaint submission, admin actions, and status updates.
3. **Data Layer (Database)** ○ Uses **MySQL** to store all user, complaint, and admin reply information. ○ Ensures data integrity, security, and fast retrieval.

2. Flow of Data

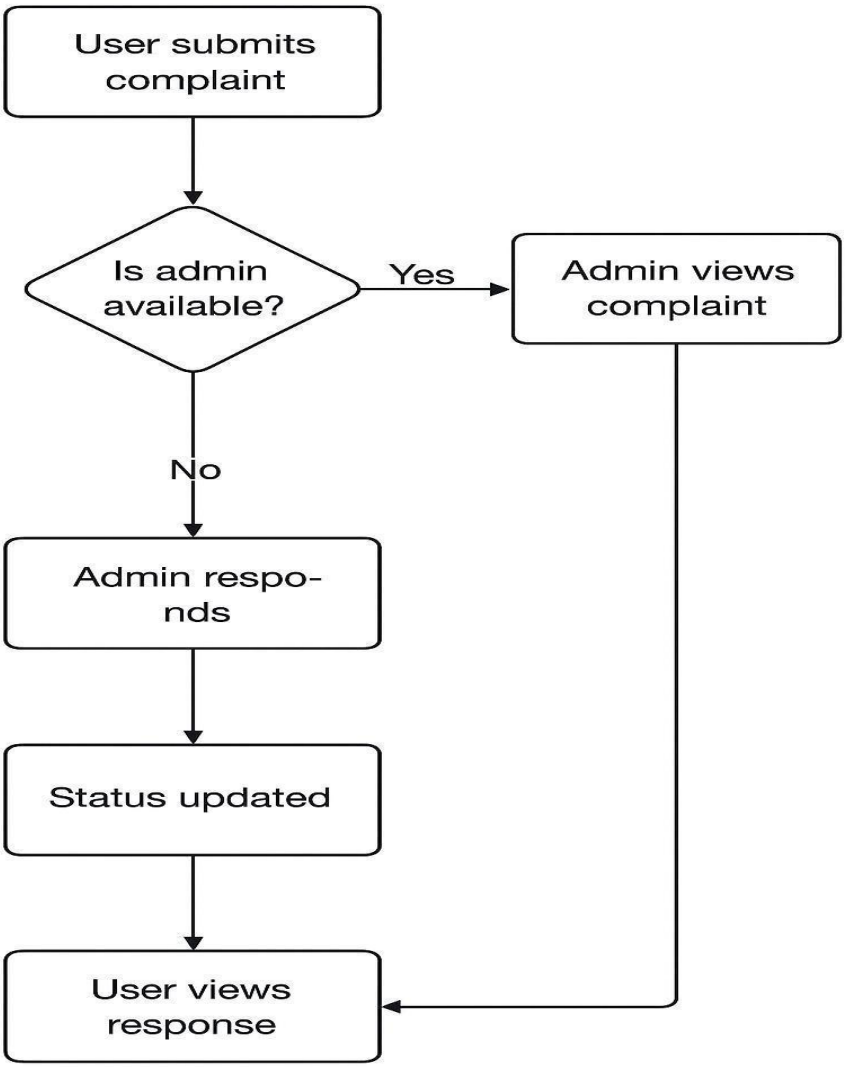
The system handles data in the following sequence: 1.

User Login / Registration:

○ Users register or log in. Credentials are verified against the database.

2. **Submit Complaint:**
   ○ Users submit complaints through a form. Data is saved in the `complaints` table.
3. **Admin Review & Reply:**
   ○ Admin accesses complaints, reviews them, and provides a reply or solution. Status is updated in the database.
4. **Status Update:**
   ○ Users can view the reply and status (resolved/pending) on their dashboard.

Flowchart


User submits complaint
Is admin available?
Yes
Admin views complaint
No
Admin respo-nds
Status updated
User views response

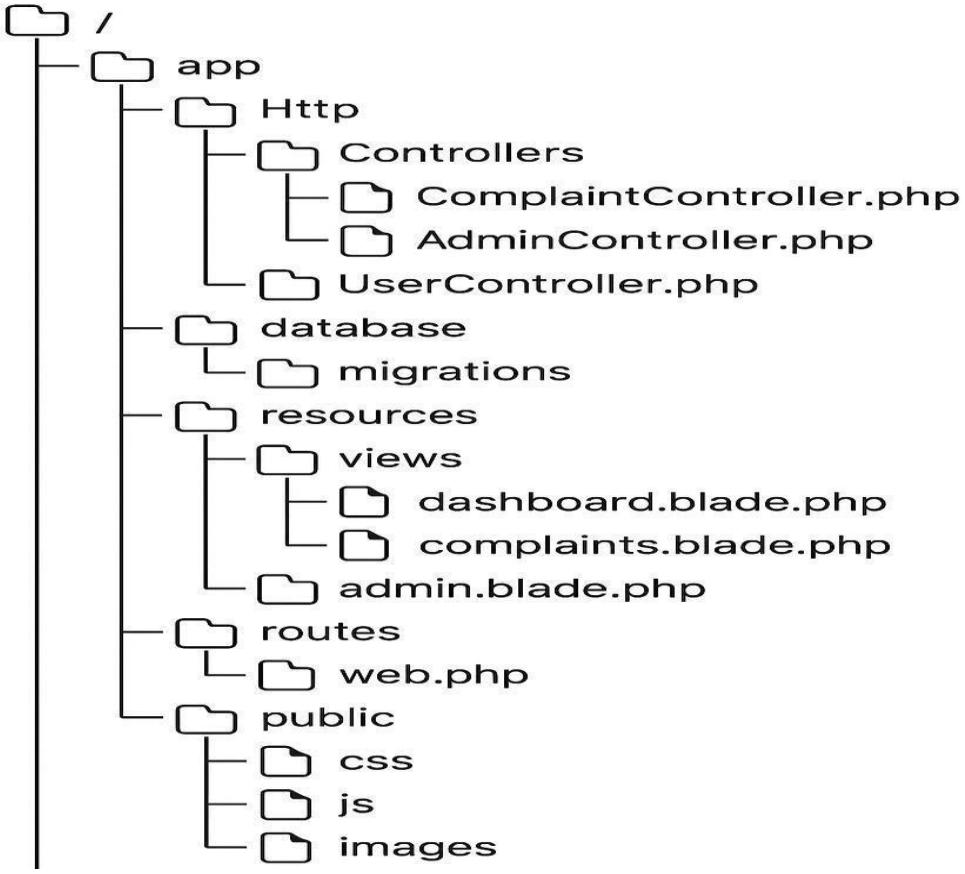
Implementation

1. Main Program Structure

The Complaint Management System follows the **MVC (Model-View-Controller)** architecture provided by the **Laravel Framework**.
This structure separates the application logic into three main parts:

- **Model:** Handles database operations such as storing, retrieving, and updating complaints and user data.
- **View:** Contains the front-end files (Blade templates) that display the web pages to users and administrators.
- **Controller:** Acts as a link between the Model and View, processing user requests and returning appropriate responses.

2. Project Directory Structure:



3.Explanation of Modules

a) User Module

- Allows users to **register, log in, and submit complaints**.
- Users can **view complaint history** and **track the status** (Pending, In Progress, Resolved).
- Example functions: storeComplaint(), viewStatus().

b) Admin Module

- Admins can **view all complaints**, **update their status**, and **reply** to users.
- Includes dashboard statistics such as total, pending, and resolved complaints.
- Example functions: updateStatus(), adminDashboard().

c) Authentication Module

- Handles **user registration, login, and role-based access** using Laravel's built-in authentication system.
- Ensures secure access to user and admin functionalities.

d) Complaint Management Module

- Enables **complaint submission**, **reply handling**, and **status tracking**.
- Complaints are stored in a MySQL database with unique IDs for easy retrieval.

e) Dashboard Module

- Displays **summary statistics** using charts and counters for admin convenience.
- Provides quick access to unresolved and resolved complaints.

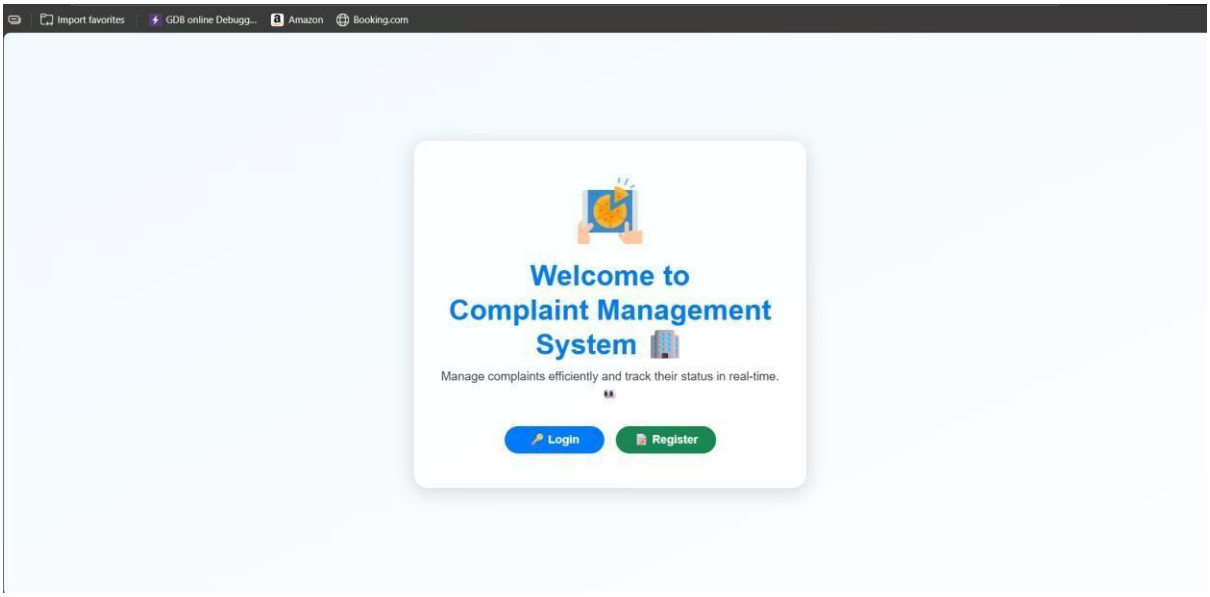
Source Code:

```
<?php namespace App\Http\Controllers; use Illuminate\Http\Request; use
App\Models\Complaint; use Illuminate\Support\Facades\Auth; class

ComplaintController extends Controller
{     public function index()
    {
        $complaints = Auth::user()->complaints()->latest()->get();         return view('complaints.index',
compact('complaints'));
    }     public function mycomplaints()
    {
        $complaints = Auth::user()->complaints()->latest()->get();         return

view('complaints.mycomplaints', compact('complaints'));
    }     public function create()
    {         return view('complaints.create');
    }     public function store(Request $request)
    {
        $request->validate([
            'title' => 'required|string|max:255',
            'description' => 'required|string',
        ]);
        Complaint::create([
            'user_id' => Auth::id(),
            'title' => $request->title,
            'description' => $request->description,
            'status' => 'pending', // default status
        ]);         return redirect()->route('complaints.index')
                        ->with('success', 'Complaint submitted successfully!');
    }     public function show(Complaint $complaint)
    {
        $this->authorizeComplaint($complaint);         return view('complaints.show', compact('complaint'));
    }     public function edit(Complaint $complaint)
    {
        $this->authorizeComplaint($complaint);         return view('complaints.edit', compact('complaint'));
}}
```

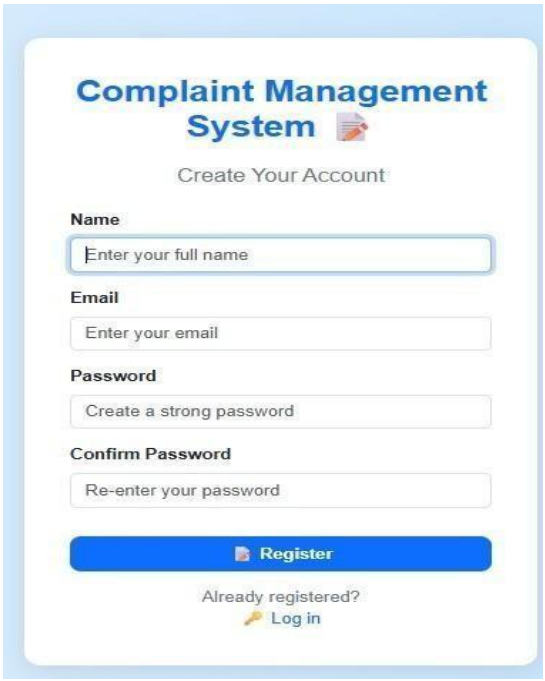
```php
<?php use Illuminate\Support\Facades\Route; use
App\Http\Controllers\ComplaintController; use
App\Http\Controllers\ProfileController; use
App\Http\Controllers\AdminController; use

App\Http\Controllers\AdminComplaintController;
Route::get('/', fn() => view('welcome'))->name('welcome');
require __DIR__.'/auth.php';
Route::middleware(['auth'])->group(function () {
    Route::get('/dashboard', fn() => redirect()->route('complaints.index'))->name('dashboard');
    Route::resource('complaints', ComplaintController::class);
    Route::get('/my-complaints', [ComplaintController::class, 'myComplaints'])->name('complaints.my');
    Route::prefix('profile')->name('profile.')->group(function () {
        Route::get('/', [ProfileController::class, 'show'])->name('show');
        Route::get('/edit', [ProfileController::class, 'edit'])->name('edit');
        Route::patch('/update', [ProfileController::class, 'update'])->name('update');
        Route::delete('/delete', [ProfileController::class, 'destroy'])->name('destroy');
    });
});
Route::middleware(['auth', 'admin'])->prefix('admin')->name('admin.')->group(function () {
    Route::get('/dashboard', [AdminController::class, 'dashboard'])->name('dashboard');
    Route::get('/users', [AdminController::class, 'users'])->name('users.index');
    Route::get('/users/{id}', [AdminController::class, 'showUser'])->name('users.show');
    Route::delete('/users/{id}', [AdminController::class, 'deleteUser'])->name('users.destroy');
    Route::resource('complaints', AdminComplaintController::class)->except(['create', 'store']);
    Route::get('complaints/{complaint}/reply', [AdminComplaintController::class, 'reply'])->name('complaints.reply');

    Route::post('complaints/{complaint}/reply', [AdminComplaintController::class, 'sendReply'])-
>name('complaints.sendReply');
});
```

Screenshots of Results:

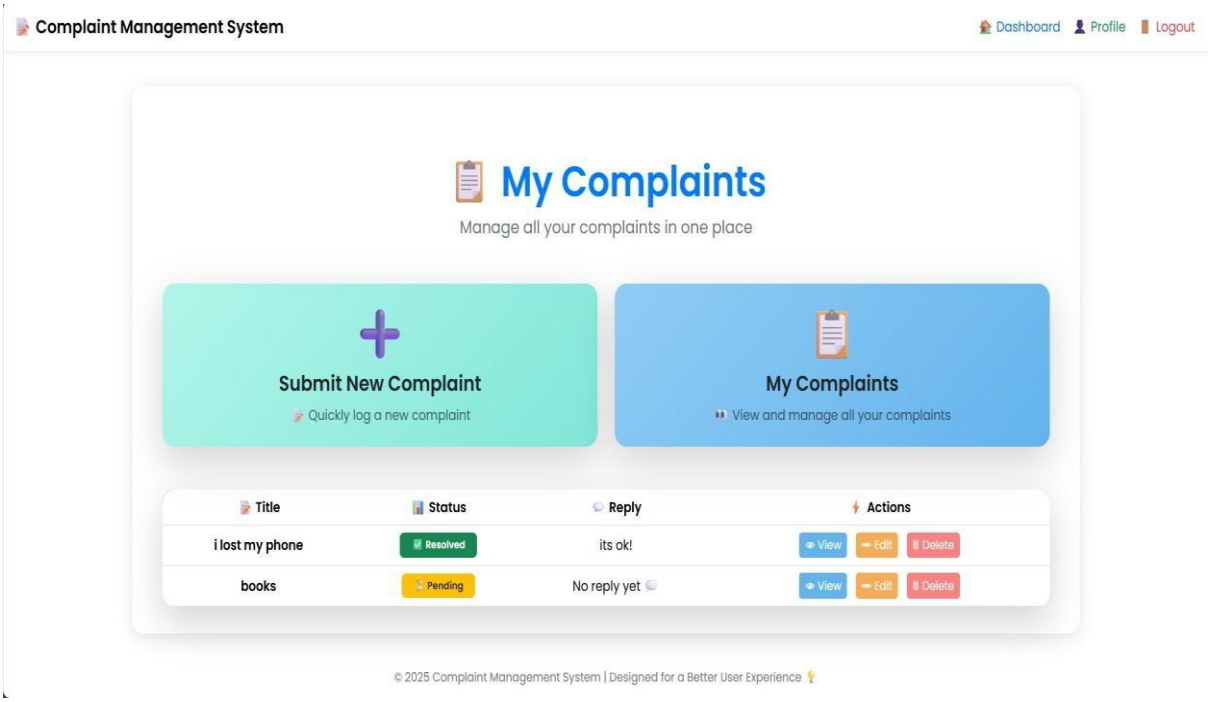
1.Welcome page:



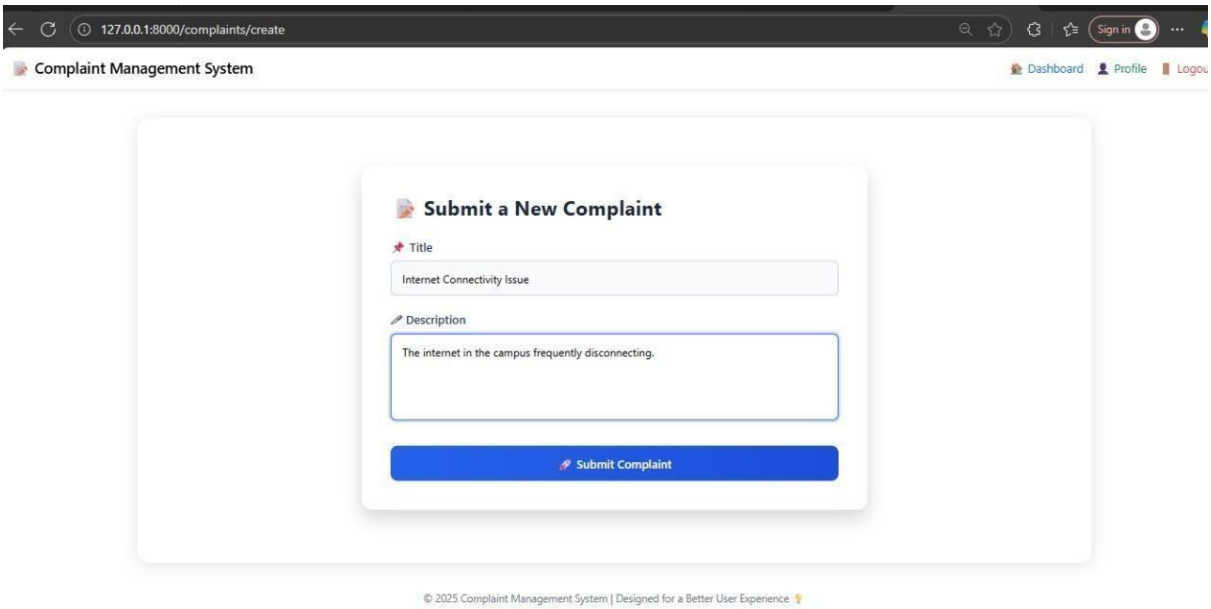
2. User Login/Registration:



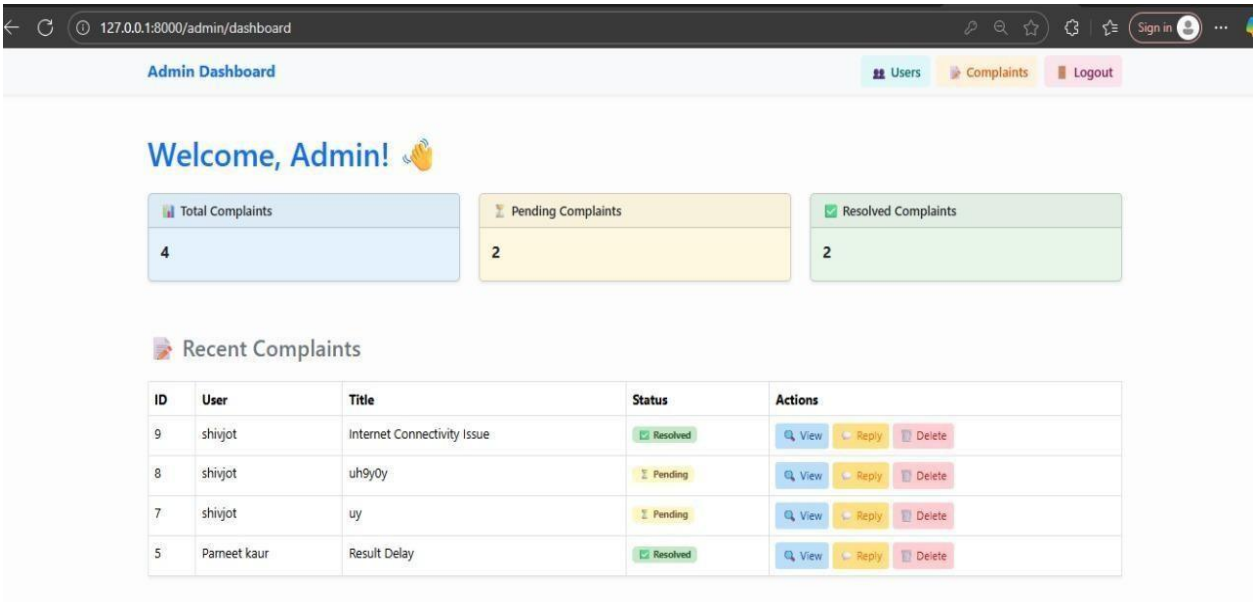
3.User Dashboard:



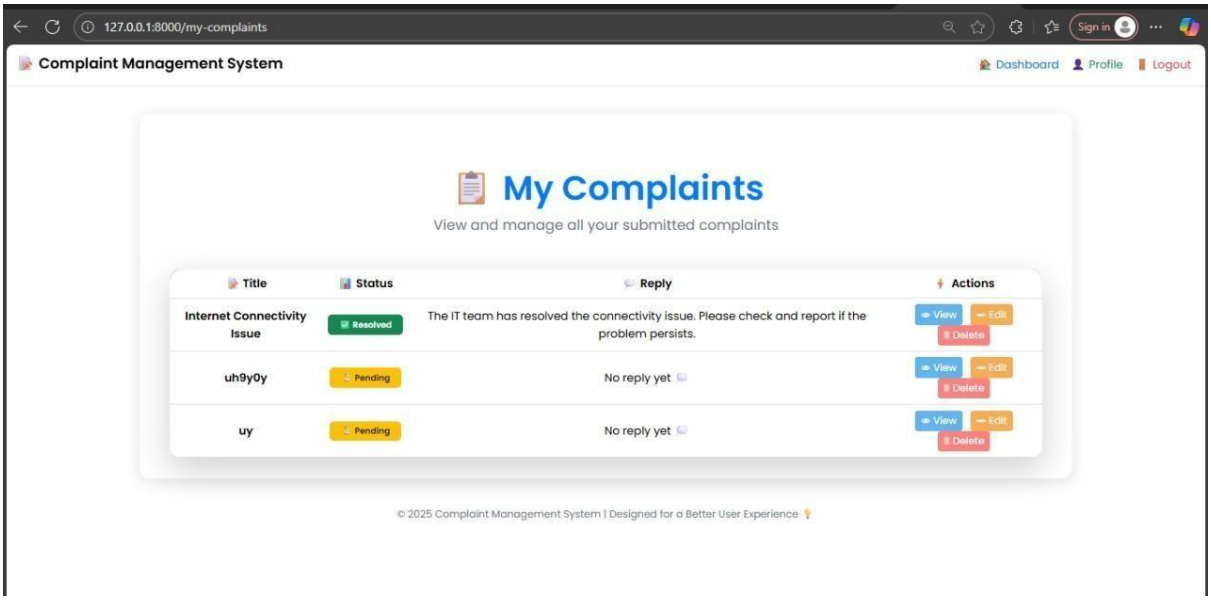
4.Complaint Submission:



5.Admin Dashboard:



6.Complaint Resolved View:



Conclusion

The Complaint Management System developed in this project provides an effective and efficient way for users to submit complaints and for administrators to manage and resolve them. The system simplifies complaint tracking, ensuring transparency and accountability, while allowing users to easily submit complaints and track their status in real-time. Administrators can efficiently manage complaints, respond promptly, and update statuses, ensuring a smooth grievance handling process. Additionally, the dashboard offers a clear overview of total, resolved, and pending complaints, enabling both users and administrators to monitor progress effectively. Overall, the system demonstrates robust functionality, ease of use, and reliability through successful testing of all modules, highlighting its value in streamlining complaint management.

References

1. **Laravel Documentation** – https://laravel.com/docs
2. **MySQL Documentation** – https://dev.mysql.com/doc/
3. **W3Schools – PHP Tutorial** – https://www.w3schools.com/php/
4. **W3Schools – HTML & CSS** – https://www.w3schools.com/
5. **GeeksforGeeks – Web Development Tutorials** - /https://www.geeksforgeeks.org/web-development/