NAAN MUDHALVAN PROJECT REPORT

Online Complaint Registration And Management System

**TEAM MEMBER**

1.BLESSY EVAGELIN D

2.DEEPAK R  
 3.DEEPAK KUMAR I

4.DHARANI G

**Project Report: Online Complaint Registration And Management System**

**1. Title**

Online Complaint Registration And Management System

## ****2. Hardware & Software Requirements****

### ****Hardware Required****:

* **Operating System**: Windows 8
* **Minimum Bandwidth**: 30 Mbps for smooth operation

### ****Software Required****:

* **Web Browsers**: Install and test using at least two different web browsers.

## ****3. Project Description****

The **Online Complaint Registration and Management System** is a platform that enables users to submit, track, and manage complaints efficiently. It aims to streamline the complaint resolution process, provide transparency, and ensure compliance with industry guidelines and regulations.

### ****Key Features****:

* **User Registration**: Create accounts to submit and monitor complaints.
* **Complaint Submission**: Enter detailed information about complaints.
* **Tracking & Notifications**: Track complaints in real-time and receive updates via email or SMS.
* **Agent Interaction**: Communicate with agents handling the complaint.
* **Assigning & Routing**: Automatically route complaints to the appropriate personnel.
* **Security & Confidentiality**: User authentication, data encryption, and access controls.

## ****4. Scenario-Based Use Case****

**Scenario**: John, a customer, encounters a defective product and decides to file a complaint through the system.

### ****Steps****:

1. **User Registration and Login**:
   * John visits the website, registers, and verifies his account via email.
   * He logs into the system using his email and password.
2. **Complaint Submission**:
   * John fills out the complaint form with details of the defective product and attaches supporting images.
   * He submits the complaint.
3. **Tracking and Notifications**:
   * John receives confirmation of his complaint registration and real-time updates.
   * Email notifications alert him to any changes in his complaint's status.
4. **Interaction with Agent**:
   * Agent Sarah is assigned to handle the complaint and communicates with John through the system.
   * They discuss the issue, and Sarah assures John of prompt resolution.
5. **Resolution and Feedback**:
   * The company offers John a replacement or refund.
   * John provides feedback on his experience, expressing satisfaction with the prompt handling of the issue.
6. **Admin Management**:
   * The admin oversees complaint distribution and ensures compliance with platform policies.

## ****5. Technical Architecture****

The system uses a **client-server model** with a focus on MERN (MongoDB, Express.js, React, Node.js) stack.

### ****5.1 Frontend****:

* **Framework**: React.js
* **Libraries**: Bootstrap and Material UI for design, Axios for API requests.
* **Features**:
  + Real-time updates on complaint status.
  + User-friendly UI components for form submissions and dashboards.

**Frontend Code Snippet (Axios API Request)**:

import axios from 'axios';

const submitComplaint = async (complaintData) => {

try {

const response = await axios.post('/api/complaints', complaintData);

console.log('Complaint submitted successfully:', response.data);

} catch (error) {

console.error('Error submitting complaint:', error);

}

};

### ****5.2 Backend****:

* **Framework**: Express.js
* **Key Functionalities**:
  + RESTful APIs for user registration, complaint submission, and tracking.
  + User authentication using JSON Web Tokens (JWT).
  + Secure communication and data handling.

**Backend Code Snippet (Complaint Submission Endpoint)**:

const express = require('express');

const router = express.Router();

router.post('/api/complaints', (req, res) => {

const { userId, description, attachments } = req.body;

// Logic to save complaint to the database

res.status(200).json({ message: 'Complaint submitted successfully' });

});

module.exports = router;

### ****5.3 Database****:

* **Database**: MongoDB
* **Schema Design**:
  + **User Collection**: Stores user information, including registration details and profiles.
  + **Complaint Collection**: Stores complaint details, tracking information, and attachments.

**MongoDB Schema Example**:

const mongoose = require('mongoose');

const complaintSchema = new mongoose.Schema({

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

description: { type: String, required: true },

status: { type: String, default: 'Pending' },

attachments: [String],

createdAt: { type: Date, default: Date.now }

});

module.exports = mongoose.model('Complaint', complaintSchema);

### ****5.4 Real-Time Communication****:

* **WebSocket & Socket.io**: For live chat between users and agents.
* **WebRTC API**: Utilized for any audio/video communication requirements.

## ****6. Security & Compliance****

* **User Authentication**: JWT-based secure login and session management.
* **Data Encryption**: AES encryption for sensitive data.
* **Access Control**: Role-based access control (RBAC) for agents and admins.
* **Compliance**: Adherence to GDPR and other relevant data protection regulations.

## ****7. Deployment & Maintenance****

### ****7.1 Deployment Strategy****:

* **Cloud Provider**: AWS or Google Cloud for scalable hosting.
* **Load Balancer**: Use NGINX for load balancing and efficient resource allocation.
* **Containers**: Docker for isolated environment setup.
* **Monitoring**: Use Prometheus and Grafana for system monitoring.

### ****7.2 Maintenance****:

* **Automated Backups**: Regular database backups for data safety.
* **Logs**: Store error and user activity logs in a centralized location.
* **Scalability**: Utilize MongoDB sharding and Express.js for horizontal scaling.

## ****8. Conclusion****

The **Online Complaint Registration And Management System** is designed to optimize the complaint handling process while ensuring a secure, user-friendly experience. It focuses on resolving complaints efficiently, enhancing customer satisfaction, and maintaining data security.