1.Name – Shubham Sharma

2. Sec - C.S.E(AI and ML)E

3. Erp - 6604633

4. Uni Roll No - 301310923156

5. Project Guide – Shreyash khedkdar

6. Project name – complaint management system

7.Github Link -

## Table of Contents

1. Introduction
2. Objectives
3. System Requirements
4. System Design
5. Implementation
6. Testing
7. Conclusion
8. Future Enhancements

## 1. Introduction

A Complaint Management System (CMS) is a software application designed to manage and resolve customer complaints efficiently. It serves as a centralized platform for recording, tracking, and analyzing complaints, ensuring that customer issues are addressed promptly and effectively. This report outlines the development and implementation of a CMS aimed at improving customer satisfaction and operational efficiency.

## 2. Objectives

The primary objectives of the Complaint Management System are:

* To provide a user-friendly interface for customers to submit complaints.
* To automate the complaint tracking process.
* To enable efficient communication between customers and support staff.
* To generate reports for analysis and decision-making.

## 3. System Requirements

### 3.1 Functional Requirements

* User registration and login.
* Complaint submission form.
* Complaint tracking and status updates.
* Admin dashboard for managing complaints.
* Reporting tools for analyzing complaint data.

### 3.2 Non-Functional Requirements

* Security: Ensure data protection and user privacy.
* Usability: Provide an intuitive interface for users.
* Performance: The system should handle multiple complaints simultaneously without lag.

## 4. System Design

### 4.1 Architecture

The CMS is designed using a client-server architecture. The client-side is a web application that allows users to interact with the system, while the server-side handles data processing and storage.

### 4.2 Database Design

The database consists of the following tables:

* **Users**: Stores user information (ID, name, email, password).
* **Complaints**: Stores complaint details (ID, user ID, description, status, timestamp).
* **Responses**: Stores responses from support staff (ID, complaint ID, response text, timestamp).

NewShare

New Chat

History

Image To App

Agents

Create Agent

Memories

SupportBlog

ProjectsMac Desktop AppAgent IDEDocsBookmarks

make a project report on complaint management system

**BLACKBOXAI**

# Project Report on Complaint Management System

## Table of Contents

1. Introduction
2. Objectives
3. System Requirements
4. System Design
5. Implementation
6. Testing
7. Conclusion
8. Future Enhancements

## 1. Introduction

A Complaint Management System (CMS) is a software application designed to manage and resolve customer complaints efficiently. It serves as a centralized platform for recording, tracking, and analyzing complaints, ensuring that customer issues are addressed promptly and effectively. This report outlines the development and implementation of a CMS aimed at improving customer satisfaction and operational efficiency.

## 2. Objectives

The primary objectives of the Complaint Management System are:

* To provide a user-friendly interface for customers to submit complaints.
* To automate the complaint tracking process.
* To enable efficient communication between customers and support staff.
* To generate reports for analysis and decision-making.
* To enhance customer satisfaction by resolving issues in a timely manner.

## 3. System Requirements

### 3.1 Functional Requirements

* User registration and login.
* Complaint submission form.
* Complaint tracking and status updates.
* Admin dashboard for managing complaints.
* Reporting tools for analyzing complaint data.

### 3.2 Non-Functional Requirements

* Security: Ensure data protection and user privacy.
* Usability: Provide an intuitive interface for users.
* Performance: The system should handle multiple complaints simultaneously without lag.
* Scalability: The system should be able to accommodate an increasing number of users and complaints.

## 4. System Design

### 4.1 Architecture

The CMS is designed using a client-server architecture. The client-side is a web application that allows users to interact with the system, while the server-side handles data processing and storage.

### 4.2 Database Design

The database consists of the following tables:

* **Users**: Stores user information (ID, name, email, password).
* **Complaints**: Stores complaint details (ID, user ID, description, status, timestamp).
* **Responses**: Stores responses from support staff (ID, complaint ID, response text, timestamp).

### 4.3 User Interface

The user interface is designed to be simple and intuitive, featuring:

* A homepage with options to register or log in.
* A complaint submission form with fields for description and category.
* A dashboard for users to view the status of their complaints.

NewShare

New Chat

History

Image To App

Agents

Create Agent

Memories

SupportBlog

ProjectsMac Desktop AppAgent IDEDocsBookmarks

make a project report on complaint management system

**BLACKBOXAI**

# Project Report on Complaint Management System

## Table of Contents

1. Introduction
2. Objectives
3. System Requirements
4. System Design
5. Implementation
6. Testing
7. Conclusion
8. Future Enhancements

## 1. Introduction

A Complaint Management System (CMS) is a software application designed to manage and resolve customer complaints efficiently. It serves as a centralized platform for recording, tracking, and analyzing complaints, ensuring that customer issues are addressed promptly and effectively. This report outlines the development and implementation of a CMS aimed at improving customer satisfaction and operational efficiency.

## 2. Objectives

The primary objectives of the Complaint Management System are:

* To provide a user-friendly interface for customers to submit complaints.
* To automate the complaint tracking process.
* To enable efficient communication between customers and support staff.
* To generate reports for analysis and decision-making.
* To enhance customer satisfaction by resolving issues in a timely manner.

## 3. System Requirements

### 3.1 Functional Requirements

* User registration and login.
* Complaint submission form.
* Complaint tracking and status updates.
* Admin dashboard for managing complaints.
* Reporting tools for analyzing complaint data.

### 3.2 Non-Functional Requirements

* Security: Ensure data protection and user privacy.
* Usability: Provide an intuitive interface for users.
* Performance: The system should handle multiple complaints simultaneously without lag.
* Scalability: The system should be able to accommodate an increasing number of users and complaints.

## 4. System Design

### 4.1 Architecture

The CMS is designed using a client-server architecture. The client-side is a web application that allows users to interact with the system, while the server-side handles data processing and storage.

### 4.2 Database Design

The database consists of the following tables:

* **Users**: Stores user information (ID, name, email, password).
* **Complaints**: Stores complaint details (ID, user ID, description, status, timestamp).
* **Responses**: Stores responses from support staff (ID, complaint ID, response text, timestamp).

### 4.3 User Interface

The user interface is designed to be simple and intuitive, featuring:

* A homepage with options to register or log in.
* A complaint submission form with fields for description and category.
* A dashboard for users to view the status of their complaints.

## 5. Implementation

The CMS is developed using the following technologies:

* **Frontend**: HTML, CSS, JavaScript (React.js)
* **Backend**: Node.js with Express.js
* **Database**: MongoDB

The implementation process involved setting up the development environment, coding the frontend and backend components, and integrating the database.

## 6. Testing

Testing was conducted to ensure the system functions as intended. The following types of testing were performed:

* **Unit Testing**: Individual components were tested for functionality.
* **Integration Testing**: The interaction between frontend and backend was tested.
* **User Acceptance Testing**: End-users tested the system to ensure it meets their needs

## 7. Conclusion

The Complaint Management System successfully addresses the need for an efficient way to manage customer complaints. By automating the complaint process and providing a user-friendly interface, the system enhances customer satisfaction and operational efficiency.

## 8. Future Enhancements

Future enhancements for the CMS may include:

* Integration with social media platforms for complaint submission.
* Implementation of a chatbot for instant responses.
* Advanced analytics for better decision-making.
* Mobile application development for on-the-go access.