**Project Title:** RESOIVENOW:YOUR PLATFORM FOR ONLINE COMPLAINTS

**Team Information**

* Team ID: LTVIP2025TMID57494
* Team Size: 4 members
* Team Leader:  Madhavi Puchabothula
* Team Members: Kancharla Rahul ,Vasthalya Tallapureddy ,Gayam Sirisha

**Table of Contents**

1. Project Overview
2. System Architecture
3. Technology Stack
4. Project Structure
5. Implementation Details
6. Development Workflow
7. Setup and Installation
8. Features and Functionality
9. API Documentation
10. Screenshots and Results
11. Challenges and Solutions
12. Future Enhancements
13. Conclusion

**1. Project Overview**

**The Online Complaints Platform is a digital system designed to help users easily file and track complaints related to public or private services. It allows users to register, submit detailed complaints, receive updates, and provide feedback on resolutions. Administrators and departments can manage and resolve issues efficiently through a central dashboard. The platform improves transparency, speeds up the complaint process, and ensures better accountability and user satisfaction.**

**2. System Architecture**

**Simple System Architecture – Online Complaints Platform**

1. **User Side (Frontend):**
   * **Website or mobile app**
   * **Users can register, log in, file complaints, check status**
2. **Server Side (Backend):**
   * **Processes user requests**
   * **Handles login, complaint submission, status updates, feedback**
3. **Database:**
   * **Stores all data**
   * **Keeps user info, complaint details, updates, feedback**
4. **Admin Panel:**
   * **For admin and department users**
   * **They can view complaints, assign, respond, close cases**
5. **Notification System:**
   * **Sends email or SMS alerts when the status of a complaint changes**
6. **Hosting Server (Cloud):**
   * **Runs the website or app online**
   * **Ensures it's always available, fast, and secure**

**3. Technology Stack**

**1. Frontend (What users see):**

* **HTML, CSS, JavaScript – to build the website**
* **React or Angular – to make the site fast and interactive**

**2. Backend (How it works behind the scenes):**

* **Node.js or Python (Django/Flask) – to handle user logins, complaints, and responses**

**3. Database (Where data is stored):**

* **MySQL or MongoDB – to save user details and complaints**

**4. Notifications:**

* **Email/SMS tools like SendGrid or Twilio – to send updates to users**

**5. Hosting (Where the website runs):**

* **AWS, Heroku, or Vercel – to make the site available online 24/7**

**4. Project Structure**

**Online-Complaints-Platform/**

**│**

**├── frontend/ → This is for the website users see**

**│ ├── pages/ → Pages like login, register, file complaint**

**│ ├── components/ → Small parts like buttons, forms**

**│ └── App.js → Main file that connects everything**

**│**

**├── backend/ → This is where the logic happens**

**│ ├── routes/ → Handles paths like /login, /complaint**

**│ ├── models/ → Defines data like User and Complaint**

**│ └── server.js → Starts the server**

**│**

**├── database/ → Stores user and complaint data**

**│ └── db-config.js → Settings to connect with the database**

**│**

**├── notifications/ → Sends emails or SMS to users**

**│**

**├── .env → Keeps passwords and keys safe**

**├── package.json → Lists project tools and libraries**

**└─**

**5. Implementation Details**

1. **Users register and log in securely.**
2. **Users submit complaints by filling a form.**
3. **Complaints are saved in the database with a status.**
4. **Users can track their complaint progress.**
5. **Admins manage and update complaints in a dashboard.**
6. **Users get email or SMS alerts when their complaint status changes.**
7. **The system uses basic security like password protection and safe data transfer.**
8. **The app is hosted online so users can access it anytime.**

**6. Development Workflow**

1. **Plan the features.**
2. **Design the user interface.**
3. **Set up the project environment.**
4. **Build the frontend.**
5. **Build the backend.**
6. **Connect frontend and backend.**
7. **Test the system.**
8. **Deploy online.**
9. **Maintain and update regularly.Top of Form**

**Bottom of Form**

**7. Setup and Installation**

1. **Install Node.js, npm, and a database (MySQL or MongoDB).**
2. **Clone the project using git clone.**
3. **Run npm install in both backend and frontend folders.**
4. **Create a .env file with your database and API settings.**
5. **Start your database server.**
6. **Run the backend server (npm start or python run).**
7. **Run the frontend server (npm start).**
8. **Open http://localhost:3000 in your browser.Top of Form**

**Bottom of Form**

**8. Features and Functionality**

* **User signup and login**
* **Submit and track complaints**
* **Admin manages and updates complaints**
* **Email/SMS notifications**
* **User feedback after resolution**
* **Search and filter complaints**
* **Secure and responsive design**

**9. API Documentation**

**GET /Complaints**

**Description: Get list of complaints.**

**Headers:  
Authorization: Bearer <token>**

**Optional Query:**

* **status (e.g., pending, resolved)**
* **userId**
* **page**
* **limit**

**Response:**

* **200 OK with complaint data (id, category, description, status, date)**
* **401 Unauthorized if no tokenTop of Form**

**Bottom of Form**

**10. Screenshots and Results**

1. **Registration Page: User signs up successfully.**
2. **Login Page: User logs in to dashboard.**
3. **Complaint Form: User submits a complaint.**
4. **User Dashboard: User views and tracks complaints.**
5. **Admin Dashboard: Admin manages and updates complaints.**
6. **Complaint Details: Shows complaint status and remarks.**
7. **Notifications: User gets email/SMS updates.**

11.Challenge and solution

* **Challenge: Users face difficulty submitting complaints.  
  Solution: Create a simple, user-friendly complaint form.**
* **Challenge: Slow complaint processing.  
  Solution: Implement an admin dashboard for quick assignment and tracking.**
* **Challenge: Lack of updates for users.  
  Solution: Send automated email/SMS notifications on status changes.**
* **Challenge: Data security risks.  
  Solution: Use password hashing and secure connections (HTTPS).**
* **Challenge: Handling large number of complaints.  
  Solution: Use database indexing and pagination for better performance.**

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

**12. Future Enhancements**

** Mobile App for easier access**

** Live Chat Support for quick help**

** AI Chatbot to auto-answer common queries**

** Multi-language Support for wider reach**

** Complaint Analytics for better decision-making**

** Feedback Rating System to improve service quality**

** Role-Based Access for different admin levels**

** Integration with Social Media for complaint submission**

**Conclusion**

**Online Complaints Platform is a smart and efficient way to manage and resolve user complaints. It makes the process faster, more transparent, and user-friendly for both users and administrators. With features like complaint tracking, notifications, and admin control, it improves service quality and user satisfaction.**

**Git Hub Link** : [**https://github.com/Madhavipuchabothula**](https://github.com/Madhavipuchabothula)

**Project code link:https://colab.research.google.com/drive/1qBKVc7BJQ0SpZCgKNR4u4q9syof-GPfp?usp=drive\_link**