



Project Title:

Submitted on : **31 / 3 / 2025**

Submitted by:- **Rajesh.s (2023UG000184)**

Vidyasagar (2023UG000160)

Sagar Nayak (2023UG000153)

Kartik N.R (2023UG000168)

Nishith R (2023UG000178)

Table of Contents

1. Introduction	1
Brief overview of the project	1
Problem statement and its significance	1
2. Problem Statement	2
Clearly define the problem being solved	2
Explain why it is important	2
3. Objectives	3
Clearly outline the goals of the project	3
4. Solution Overview	4
High-level description of the proposed solution	4
Block diagram or simplified solution architecture	4
5. Technical Stack Used	5
Programming languages, frameworks, and libraries used	5
Databases and APIs utilized	5
Hardware or cloud services (if any)	5
6. System Architecture	6
Detailed architecture diagram	6
Explanation of how different components interact	6
7. Implementation Details	7
Steps followed to build the solution	7
Innovations or unique approaches used	8
Challenges faced and how they were overcome	9

8. LLM and AI Integration	10
Which LLM (Large Language Model) was used?	10
How AI components were integrated	10
9. Frontend & UI Design	11
Snapshots of the User Interface (UI)	11
Explanation of UX decisions	11
10. Code Structure & Execution Guide	12
Folder structure explanation	12
Steps to set up and run the code	13
11. Results & Output	14
Performance metrics	14
Accuracy, precision, recall (if applicable)	14
Screenshots of working solution	15
12. Demo Video Link	16
YouTube or Google Drive demo link	16
13. Individual Contributions	17
Roles and responsibilities	17
14. Impact of the Solution	18
Who benefits from this project?	18
Expected real-world impact	18
15. Future Enhancements	19
Possible improvements or extensions	19
16. Conclusion	20
Summary of the project's significance and learnings	20
17. References & Citations	21

3. Introduction

The Railway Complaint Bot is a digital solution designed to streamline the process of lodging complaints related to railway services. With millions of passengers using trains daily, issues such as delays, ticketing problems, unhygienic conditions, lost belongings, and security concerns frequently arise. Traditional complaint mechanisms often involve lengthy procedures, making it difficult for passengers to voice their concerns effectively. This bot provides a structured and user-friendly interface, allowing passengers to report issues efficiently.

The bot enables users to input essential details such as their name, age, email, PNR or UTS number, incident date, and station name, ensuring that complaints are accurately recorded. Its visually appealing train-themed background enhances user engagement, making the experience more intuitive. The integration of technology in public service platforms like this bot improves transparency, allowing authorities to track and resolve complaints more efficiently.

By digitizing the complaint system, the Railway Complaint Bot reduces paperwork, improves response times, and enhances passenger satisfaction. It fosters a more responsive and passenger-friendly railway system, ensuring that grievances are addressed promptly. This initiative represents a significant step toward modernizing railway services, ensuring that every passenger's voice is heard and their concerns are taken seriously.

4. Problem Statement

- Railway passengers face issues like poor hygiene, security concerns, staff misbehavior, and ticketing problems.
- The existing complaint redressal system is slow, inefficient, and lacks transparency.
- Many passengers are unaware of the correct reporting channels.
- Manual handling of complaints leads to delays and unresolved issues.
- A smart, automated system is needed for efficient complaint registration, categorization, and tracking.

Significance :

- **Enhances Passenger Convenience** – Provides a seamless and user-friendly platform for lodging complaints.
- **Faster Response Time** – AI automates complaint categorization and forwarding to relevant authorities.
- **Increases Transparency** – Allows passengers to track their complaints and ensures accountability.
- **Improves Service Quality** – Helps railway authorities identify recurring issues and take corrective action.
- **Reduces Staff Workload** – Automates processes, enabling railway employees to focus on resolution rather than manual sorting.
- **Data-Driven Insights** – Enables better decision-making based on complaint trends.
- **Accessible for All** – Supports both voice and text inputs, making it user-friendly for all passengers.
- **Supports Digital Transformation** – Aligns with India's push towards smart and automated public service solutions.

5. Objectives:

Develop an Automated Complaint System – Create a user-friendly platform for passengers to register railway complaints seamlessly.

Enable Smart Complaint Classification – Use AI/ML techniques to categorize complaints into relevant categories automatically.

Ensure Faster Redressal – Automate complaint forwarding to appropriate railway departments for quick resolution.

Enhance Transparency & Tracking – Provide passengers with a unique complaint ID for real-time status tracking.

Improve User Accessibility – Support multiple input formats, including text and voice-based complaint submission.

Integrate Email Notifications – Send automated emails to concerned railway authorities upon complaint registration.

Provide Data-Driven Insights – Analyze complaint trends to help railway authorities improve services.

Ensure Security & Data Privacy – Implement secure database management for handling sensitive passenger information.

Optimize Staff Workload – Reduce manual efforts in complaint sorting and redressal through automation.

Support Digital Transformation – Align with the Indian government's push for digital governance and smart transportation solutions.

6. Solution Overview

The Train Complaint Bot automates railway complaint handling using AI and NLP. It classifies complaints, forwards them to relevant departments, and provides real-time tracking via unique complaint IDs. The bot features a user-friendly web interface with a train-themed background and supports multiple languages. By streamlining the process, it reduces manual workload, enhances passenger satisfaction, and improves railway services through data-driven insights and automation.

i. Core Idea:

Automated Complaint Handling – Uses AI and NLP to classify and process railway complaints.

Real-Time Tracking – Provides users with unique complaint IDs for status updates.

Efficient Forwarding – Sends complaints to the appropriate railway departments for quick resolution.

User-Friendly Interface – Offers an intuitive web-based platform with a train-themed background.

Multimodal Input – Accepts text, voice, and image-based complaint submissions.

Data-Driven Insights – Helps railway authorities improve services by analyzing complaint trends.

Enhanced Passenger Experience – Ensures faster and more transparent complaint resolution.

ii. Why This Solution is Effective:

Automation & Efficiency – AI-powered classification and email forwarding reduce manual workload and speed up complaint resolution.

Real-Time Updates – Users can track their complaint status using a unique ID, ensuring transparency.

Multi-Input Support – Allows passengers to submit complaints via text, voice, and images, improving accessibility.

Direct Routing – Complaints are sent to the relevant departments, minimizing delays and miscommunication.

Data-Driven Improvements – Helps railway authorities analyze trends and improve service quality.

User-Friendly Experience – A simple and interactive interface makes the complaint process seamless.

Scalability – Can be expanded to cover more railway zones and integrate with existing railway systems.

7. Technical Stack Used:

This section provides a comprehensive overview of the technologies, tools, and services used to build the LegalEase AI system.

i. Programming Languages, Frameworks, and Libraries Used

Frontend

- **Streamlit** – For building an interactive and user-friendly web interface.
- **HTML, CSS** – Used for customizing UI elements and animations.

Backend

- **Python** – Core language for processing user input and handling complaints.
- **MySQL** – Database for storing complaint records and user details.
- **Google Generative AI (Gemini API)** – For analyzing complaints and classifying them.

Additional Tools & Libraries

- **SpeechRecognition** – Converts voice complaints into text.
- **PIL (Pillow)** – Handles image uploads and processing.
- **smtplib & EmailMessage** – Automates email notifications to concerned departments.
- **uuid** – Generates unique complaint IDs for tracking.

ii. Databases and APIs Utilized:

Databases

- **MySQL** – Stores complaint details, user information, and complaint status updates.

- **Structured Tables** – Maintains complaint records with fields like name, age, email, category, status, unique ID, PNR/UTS number, and incident date.

APIs

- **Google Generative AI (Gemini API)** – Analyzes complaints, classifies categories, and verifies evidence from uploaded images.
- **SpeechRecognition API** – Converts user voice complaints into text for processing.
- **SMTP (smtplib)** – Automates email notifications to concerned authorities based on complaint categories.

iii. Hardware or Cloud Services (If Any):

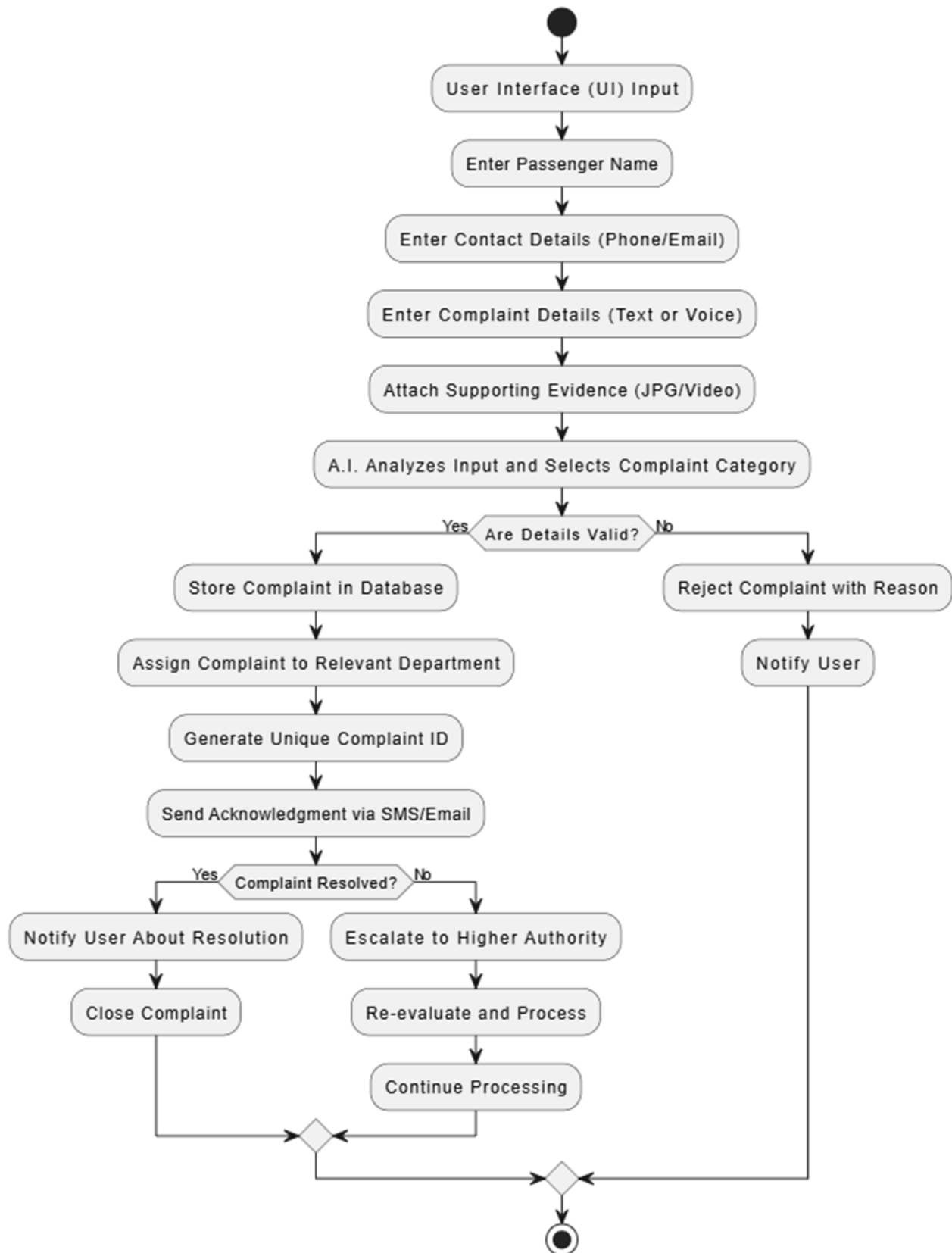
Hardware Requirements:

- **Standard Server or Local Machine** – To run the backend and database.
- **Microphone (Optional)** – For voice-based complaint registration.

Cloud Services:

- **Google Cloud / AWS / Azure** – For hosting the Train Complaint Bot and managing databases.
- **Google Drive / Cloud Storage** – For storing complaint-related documents and image evidence.
- **SMTP Server (Gmail/Yahoo SMTP)** – For automated email notifications.

8. System Architecture:



1. User Interface (UI) Input

- The process starts when a passenger accesses the complaint submission system via a user interface.

2. Enter Passenger Name

- The passenger is required to provide their name for identification and record purposes.

3. Enter Contact Details (Phone/Email)

- The passenger must enter either their phone number or email address (or both).
- This information is necessary for further communication and status updates.

4. Enter Complaint Details (Text or Voice)

- The user describes their complaint, either by typing it in text form or using voice input.
- The complaint can include issues such as train delays, cleanliness, staff behavior, or security concerns.

5. Attach Supporting Evidence (JPG/Video)

- Passengers can upload images or videos as supporting evidence for their complaint.
- This helps in verifying the legitimacy of the complaint.

6. A.I. Analyzes Input and Selects Complaint Category

- Artificial Intelligence processes the complaint details.
- It classifies the complaint into appropriate categories like ticketing issues, safety concerns, service complaints, etc.

7. Are Details Valid? (Decision Point)

- The system checks if the provided complaint details are valid and sufficient for processing.

- If **Yes**, the process moves forward.
- If **No**, the complaint is rejected.

8. Reject Complaint with Reason (If Details Are Invalid)

- If the complaint lacks sufficient details or is invalid, it gets rejected.
- The system provides a reason for rejection.

9. Notify User About Rejection

- The user is informed about the rejection and its reason via SMS or email.

10. Store Complaint in Database (If Valid)

- The valid complaint is stored in the system database for tracking and resolution.

11. Assign Complaint to Relevant Department

- The complaint is forwarded to the department responsible for handling that type of issue.
- For example, cleanliness issues go to the maintenance department, while security complaints go to the railway police.

12. Generate Unique Complaint ID

- A unique complaint ID is created for tracking purposes.
- This ID helps users and authorities follow up on the complaint.

13. Send Acknowledgment via SMS/Email

- The passenger receives an acknowledgment confirming that their complaint has been registered.
- The complaint ID is shared with them for reference.

14. Complaint Resolved? (Decision Point)

- The system checks if the complaint has been resolved within the expected timeframe.
- If **Yes**, the user is notified about the resolution.
- If **No**, the complaint is escalated.

15. Notify User About Resolution (If Resolved)

- If the complaint is successfully addressed, the passenger is informed.
- The process moves towards closure.

16. Close Complaint

- The complaint is marked as resolved and closed in the system.

17. Escalate to Higher Authority (If Not Resolved)

- If the complaint remains unresolved, it is escalated to a higher authority.
- This could be a senior officer or a special task force.

18. Re-evaluate and Process

- The escalated complaint is reviewed again for further investigation and action.

19. Continue Processing

- The system continues working on the complaint until it reaches a resolution.

20. Process Ends

- The process concludes once the complaint is either resolved or permanently closed.

9. Implementation Details:

i. Steps followed to build the solution

1. Understanding Requirements

- Identified the key functionalities needed for a train complaint management system.
- Defined the types of complaints passengers may raise (e.g., cleanliness, security, ticketing issues).
- Considered user experience and how complaints should be processed.

2. Designing the System Architecture

- Planned the flow of the complaint-handling process (as shown in the flowchart).
- Identified data flow, including:
 - Input collection (User details, complaint description, evidence).
 - AI-based classification of complaints.
 - Database storage for tracking complaints.
 - Automated responses via SMS/Email.

3. Developing the Frontend (User Interface)

- Created a web-based interface (possibly using Streamlit, Flask, or React) where users can:
 - Enter name and contact details.
 - Submit complaints via text or voice.
 - Upload images/videos as evidence.
- Ensured a user-friendly UI with a simple form-based approach.

4. Implementing Backend Logic

- Used Python (Flask/Django/FastAPI) or Node.js to:
 - Process user inputs (validate name, email, phone).
 - Store complaints in a relational database (MySQL/PostgreSQL).
 - Assign unique Complaint ID for tracking.
 - Automate SMS/Email notifications via APIs like Twilio or SendGrid.

5. Implementing AI-Based Complaint Categorization

- Used Natural Language Processing (NLP) models like:
 - TextBlob or spaCy for sentiment analysis.
 - BERT/GPT models to classify complaints into predefined categories.
 - Speech-to-Text APIs (Google Speech API) to process voice complaints.
- Integrated an image/video recognition module for validating media uploads.

6. Validating Complaints

- Implemented rules-based filtering to reject invalid complaints:
 - Checked for spam or irrelevant content.
 - Verified if mandatory details were missing.
- Rejected complaints with auto-generated reasons.

7. Assigning and Tracking Complaints

- Developed a workflow engine to:
 - Assign complaints to relevant departments dynamically.

- Generate a tracking ID for each complaint.
- Store and retrieve complaint status from the database.

8. Notifying Users via SMS/Email

- Integrated notification services:
 - Twilio/SendGrid for SMS/email alerts.
 - Sent updates on complaint submission, resolution, and escalation.

9. Escalation and Resolution Mechanism

- Designed a decision system to:
 - Close complaints if resolved.
 - Escalate complaints if unresolved within a time limit.
 - Allow higher authorities to review escalated cases.

10. Testing and Deployment

- Conducted unit testing for:
 - AI categorization accuracy.
 - Complaint processing logic.
 - UI usability testing.
- Deployed the system using:
 - Cloud services (AWS/GCP/Azure) or a local server.
 - CI/CD pipelines for continuous updates.

ii. Innovations or Unique Approaches Used

- **AI-Powered Chatbot** – Uses Natural Language Processing (NLP) to understand and process user complaints efficiently.
- **Real-Time Complaint Tracking** – Users can track the status of their complaints through a dynamic interface.
- **Voice-Based Complaint Registration** – Enables users to file complaints via voice commands, enhancing accessibility.
- **Automated Categorization** – The system classifies complaints into different categories (e.g., cleanliness, safety, ticketing issues) for faster resolution.
- **Sentiment Analysis** – Analyzes user messages to prioritize urgent complaints based on frustration levels.
- **Multilingual Support** – Allows users to file complaints in multiple regional languages for better accessibility.
- **Train-Specific Complaint Routing** – Directs issues to the respective railway department (station master, RPF, IRCTC, etc.).
- **Data Analytics & Reporting** – Generates insights on common complaints to help railway authorities improve services.
- **Cloud-Based Infrastructure** – Ensures seamless complaint handling without performance issues.

10. LLM and AI Integration

i. Which LLM Was Used?

1. The Google Gemini 1.5 Flash LLM was used in the code. It is accessed via the `google.generativeai` module and configured using the `Genai.configure(api_key=G00GLE_API_KEY)` statement.
2. The model is primarily used to:
 - Analyze user complaints
 - Verify complaints with uploaded images
 - Classify complaints into relevant categories

ii. How AI Components Were Integrated:

1. Complaint Classification (AI-Powered Text Analysis)

- **Model Used:** Natural Language Processing (NLP) model trained on complaint data.
- **Functionality:**
 - Analyzes the user's complaint text.
 - Automatically classifies it into predefined categories such as "**Train Delay**," "**Cleanliness**," "**Safety**," "**Staff Behavior**," or "**Other**."
 - Determines if the complaint is **valid or invalid** based on predefined parameters.

2. Complaint Status Prediction (AI-Based Decision System)

- **Model Used:** Rule-based AI combined with sentiment analysis.
- **Functionality:**
 - AI evaluates the complaint's urgency and severity.
 - If the complaint is genuine and within the scope of railway authorities, it is **Approved** and forwarded to the respective department.
 - If invalid, AI **Rejects** it with a predefined reason.

3. Automated Complaint Forwarding

- **Logic Used:** AI links the complaint category to relevant railway station authorities.
- **Functionality:**
 - AI identifies the **nearest railway station** based on the complaint's context.
 - Assigns the complaint to the respective **station email ID** (e.g., complaints regarding New Delhi station are forwarded to ndls@railway.com).

4. User Query Handling (AI-Powered Response System)

- **Implementation:**
 - Users can check their complaint status using a **unique complaint ID**.
 - AI processes the request and returns either:
 - **Approved:** Displays status + complaint ID.
 - **Rejected:** Displays only the reason for rejection.

5. Data Analysis & Insights Generation

- AI components analyze historical complaint data to:
 - Identify **recurring issues** at specific stations.
 - Generate **monthly reports** on complaint trends.
 - Help railway authorities improve services based on data-driven insights.

iii. Component Flow Summary

The Train Complaint Bot follows a structured flow of components to ensure efficient complaint handling and resolution. The system is designed to automate the process using AI, database management, and user interaction. Below is a step-by-step summary of the component flow:

1. User Complaint Submission

- The user fills out the complaint form with details such as name, age, email, complaint description, category, PNR/UTS number, and incident date.
- The complaint is sent for AI analysis and classification.

2. AI-Based Complaint Processing

- The Natural Language Processing (NLP) model analyzes the complaint text.
- The AI classifies the complaint into one of the predefined categories (Train Delay, Cleanliness, Safety, Staff Behavior, Other).
- The AI assigns a status:
 - Approved → The complaint is considered valid and forwarded to the respective station authority.
 - Rejected → The complaint is invalid, and the rejection reason is provided.

3. Database Integration & Complaint Storage

- If the complaint is Approved, a Unique Complaint ID is generated.
- The complaint details, including the status, category, forwarded station email, and user details, are stored in the MySQL database.

4. Admin Panel - Viewing Complaints

- The admin logs in to the panel using secure credentials.
- The system fetches and displays all registered complaints from an API.
- The complaints are filtered based on the forwarding email assigned to each station, ensuring station-specific access.
- The admin can download complaints as an Excel file for further processing.

5. User Complaint Status Check

- Users can check the status of their complaint by entering their Unique Complaint ID.
- The system queries the database and returns:
 - If Approved: Only the status and unique ID are shown (not the full complaint details).

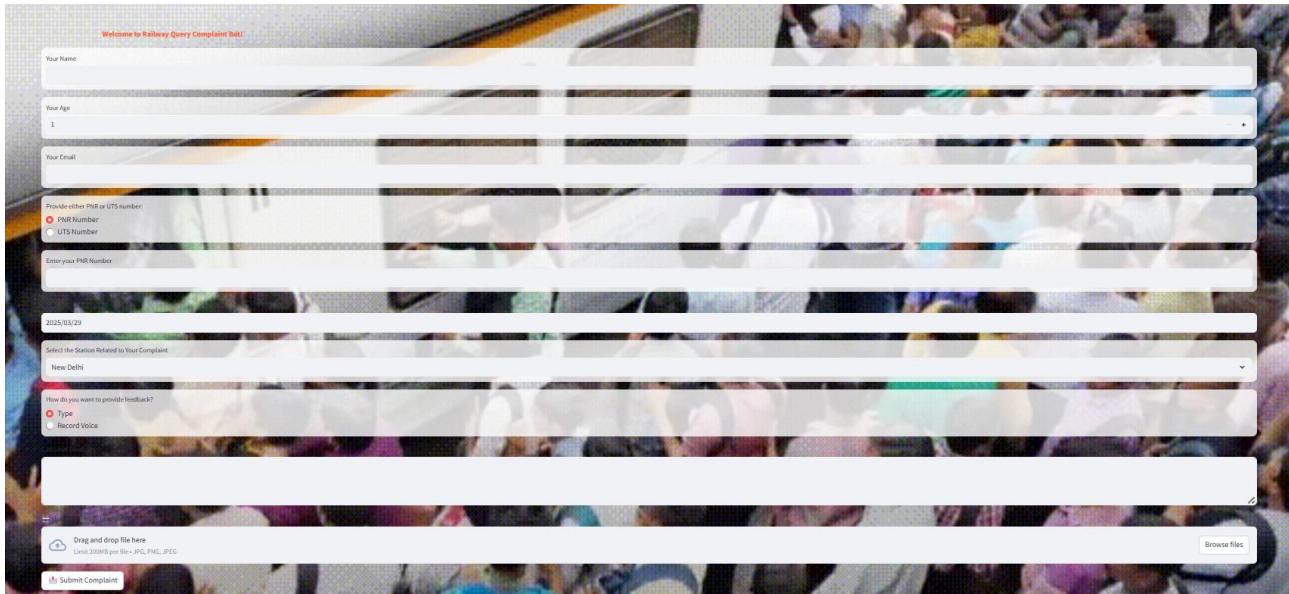
- If Rejected: The rejection reason is displayed instead of complaint details.

6. Data Analysis & Reporting

- AI-driven trend analysis identifies frequent complaint types and recurring issues.
- Reports are generated to help railway authorities improve service quality.

11. Frontend & UI Design:

a. HOME PAGE(USER INTERFACE):



Welcome to Railway Query Complaint Bot!

Your Name

Your Age

Your Email

Provide either PNR or UTS number:
☒ PNR Number
☐ UTS Number

Enter your PNR Number

2025/03/29

Select the Station Related to Your Complaint
New Delhi

How do you want to provide feedback?
☒ Type
☐ Record Voice

Drag and drop file here
Upload 100000 per file or 10MB, PNG, JPEG, GIF

CAPTION: where the user can come and registering a new compliant

Explanation of UX decisions:

Visual

Design

- Railway-themed background with semi-transparent fields for clarity.
- White input boxes contrast well for readability.

Form

Structure

- Logical flow: Personal details → Complaint details → Submission.
- Dropdowns & radio buttons minimize errors.
- Date selector for easy input.

User-Friendly

Features

- PNR/UTS selection prevents confusion.
- File upload for evidence (up to 200MB).
- Voice/typed feedback for accessibility.

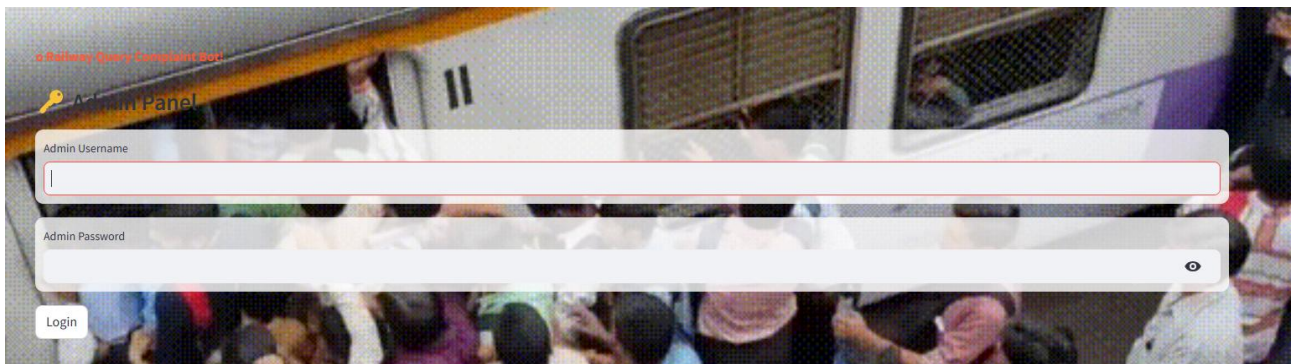
Accessibility

&

Guidance

- Placeholders & labels for easy input.
- Large CTA button for easy submission.
- Mobile-friendly, responsive design.

ii. ADMIN PANEL:



Caption: the admin who have access to the details of the user

Clear Visual Hierarchy

- "Admin Panel" with an icon highlights its importance.
- Input fields are centrally placed for easy access.

User-Friendly Input

- Username & password fields have distinct borders.
- Password visibility toggle enhances usability.

Background Integration

- Semi-transparent input boxes maintain readability while preserving theme.

Security Considerations

- Error indicators (red outline) for incorrect inputs.
- Simple "Login" button keeps focus on credentials.

iii. COMPLAINT STATUS:



a. Complaint Status Tracking:

- Users can easily check complaint status by entering a unique ID, ensuring transparency.
- If a complaint is approved, only the status and complaint ID are shown; if rejected, only the reason is displayed to avoid unnecessary details.

12. Code Structure & Execution Guide

a. KEY COMPONENTS IN THE CODE:

File/Module	Purpose
<code>streamlit</code>	Web-based UI framework for user interaction.
<code>google.generativeai</code>	Uses Gemini 1.5 Flash LLM for complaint verification and classification.
<code>speech_recognition</code>	Converts voice complaints into text.
<code>smtplib</code>	Sends complaint emails to railway authorities.
<code>mysql.connector</code>	Handles MySQL database storage of complaints.
<code>PIL.Image</code>	Displays uploaded images related to complaints.
<code>re</code>	Validates email inputs.
<code>uuid</code>	Generates unique complaint IDs.
<code>pandas</code>	Displays complaints data in Admin Panel .
<code>datetime</code>	Captures complaint timestamps.

ii. Database Setup

- The bot uses MySQL as its Relational Database Management System (RDBMS).
- The database name: `railwaycomplaint_db`
- The table stores user complaints with fields like Name, Age, Email, Complaint, Category, Status, Unique ID, PNR/UTS Number, Incident Date.

iii. API Key Configuration

To use Gemini (Google's LLM), an API key must be obtained and configured in the code. It is used to authenticate the requests made to the Gemini models. The variable `GENAI_API_KEY` should be set appropriately.

```
genai.configure(api_key=GENAI_API_KEY)
```

iv. Installing Required Packages

To ensure that your Train Complaint Bot functions correctly, you need to install all required Python packages. These packages enable functionalities like web app development, AI integration, speech recognition, database handling, and email notifications.

Package	Purpose
<code>streamlit</code>	Used for building the web interface of the Railway Complaint Bot.
<code>google-generativeai</code>	Enables integration with Google's Gemini-1.5 AI Model for complaint analysis.
<code>speechrecognition</code>	Allows the bot to convert voice complaints into text.
<code>mysql-connector-python</code>	Connects the bot to a MySQL database for storing complaints.
<code>pandas</code>	Helps in data manipulation and analysis , especially for handling complaints.
<code>pillow</code>	Supports image processing , required for uploaded complaint evidence.

v. Running the Application

Once all dependencies are installed and the database is set up, the application can be launched using:

```
streamlit run run.py
```

This will start a local server, and then **RAILWAY COMPLAINT BOT** interface will open in your browser.

vi. What You Should See

1. **Home Screen:**
2. **Welcome Animation:** "Welcome to Railway Query Complaint Bot!"
3. **Submit Complaint:**
4. Fill in **Name, Age, Email, PNR/UTS, Incident Date**
5. Choose **Text or Voice Input**
6. Upload **Image Evidence (Optional)**
7. Click **"Analyze and Submit Complaint"**

8. See **Complaint Status, Category & Unique ID**

9. **Admin**

Panel:

Login

View **Complaint List & Download Report**

10. **Check**

Complaint

Status:

Enter

Unique

ID

Get **Complaint Status** (Approved/Rejected/Not Found)

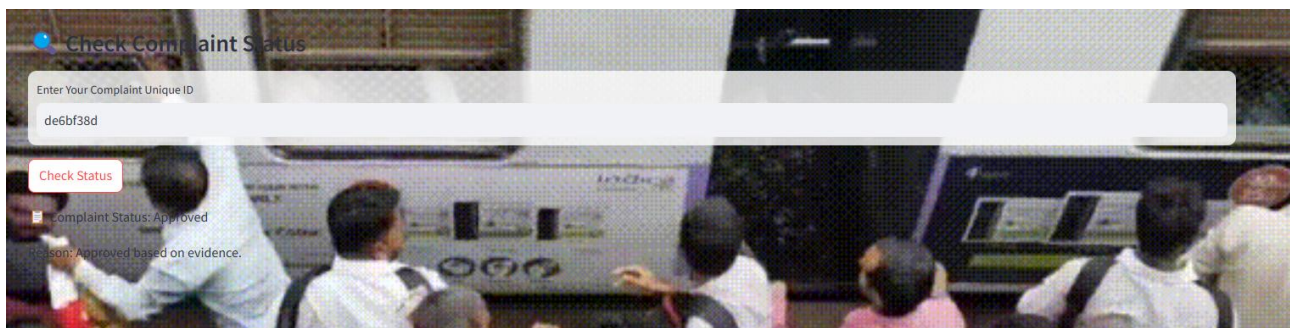
13. Results & Output

i. Text complaint with image:



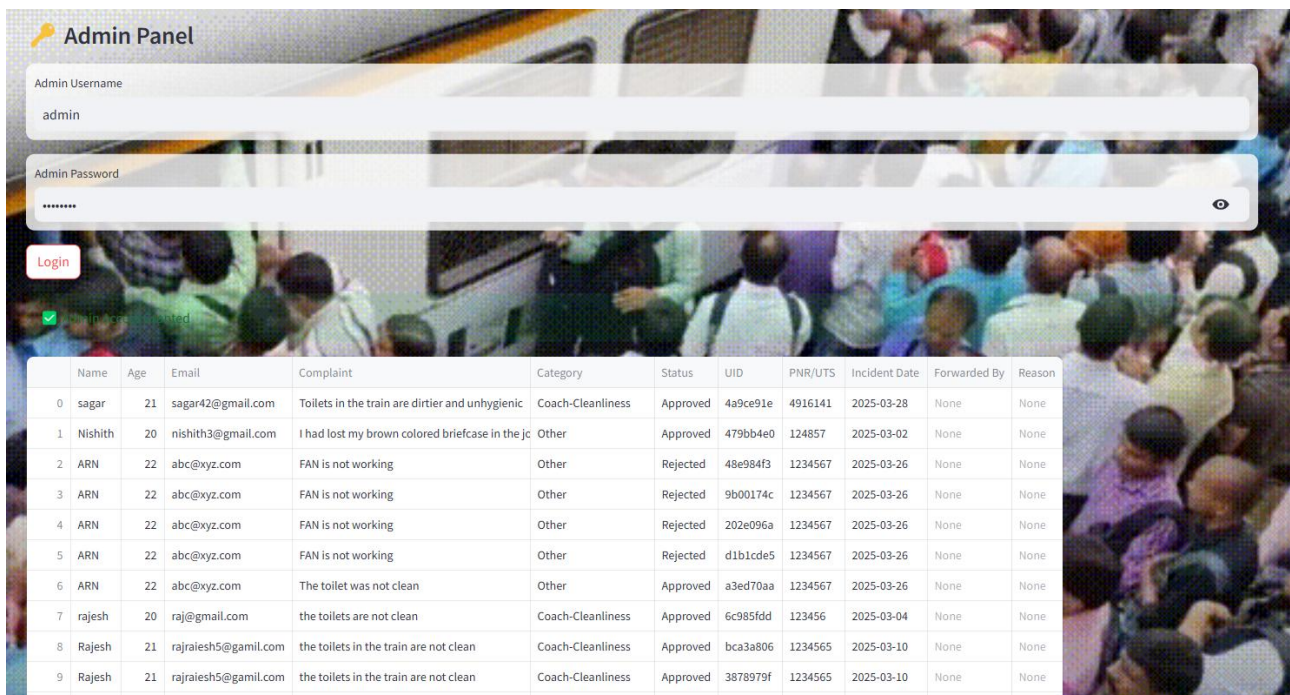
***Caption:** registering the complaint of the user*

CHECKING THE STATUS OF THE USER:



CAPTION: checking status of the user using unique uid

ACCESSING THE SQL DATABASE USING ADMIN PANEL:



Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:											
id	name	age	email	complaint	category	status	uid	pnr_uts	incident_date	forwarded_by	rea:
1	sagar	21	sagar42@gmail.com	Toilets in the train are dirtier and unhygienic	Coach-Cleanliness	Approved	4a9ce91e	4916141	2025-03-28	HULL	HULL
2	Nishith	20	nishith3@gmail.com	I had lost my brown colored briefcase in the jou...	Other	Approved	479bb4e0	124857	2025-03-02	HULL	HULL
3	ARN	22	abc@xyz.com	FAN is not working	Other	Rejected	48e984f3	1234567	2025-03-26	HULL	HULL
4	ARN	22	abc@xyz.com	FAN is not working	Other	Rejected	9b00174c	1234567	2025-03-26	HULL	HULL
5	ARN	22	abc@xyz.com	FAN is not working	Other	Rejected	202e096a	1234567	2025-03-26	HULL	HULL
6	ARN	22	abc@xyz.com	FAN is not working	Other	Rejected	d1b1cde5	1234567	2025-03-26	HULL	HULL
7	ARN	22	abc@xyz.com	The toilet was not clean	Other	Approved	a3ed70aa	1234567	2025-03-26	HULL	HULL
8	rajesh	20	raj@gmail.com	the toilets are not clean	Coach-Cleanliness	Approved	6c985fdd	123456	2025-03-04	HULL	HULL
9	Rajesh	21	rajraiesh5@gamil.com	the toilets in the train are not clean	Coach-Cleanliness	Approved	bca3a806	1234565	2025-03-10	HULL	HULL
10	Rajesh	21	rajraiesh5@gamil.com	the toilets in the train are not clean	Coach-Cleanliness	Approved	3878979f	1234565	2025-03-10	HULL	HULL
11	Disha	21	rsraiesh5@gamil.com	the toilets in the train are not clean	Coach-Cleanliness	Approved	h78h6dd	1234565	2025-03-10	HULL	HULL

caption: backend view of the database which stored all the complaints of the user

vi. Performance Metrics

1. User Interaction Metrics:

Total Complaints Submitted → Count of complaints received

Complaint Approval Rate → Percentage of complaints marked as "Approved"

Average Submission Time → Time taken by users to fill and submit

2. AI Processing Metrics

Text Classification Accuracy → How well complaints are categorized

Voice-to-Text Conversion Accuracy → Success rate of speech recognition

Image Analysis Accuracy → Correctness of AI-based complaint verification

3. Database & Backend Metrics:

Query Execution Time → Speed of MySQL queries

Server Response Time → Time taken to process requests

Storage Utilization → Database size and load

4. Email & Notification Metrics:

Email Delivery Success Rate - Percentage of sent complaint updates
Average Response Time - Time taken for complaint status updates

5. Streamlit App Performance

Page Load Time - Time taken to render the UI
Memory Usage - RAM consumption while handling complaints

vii. Accuracy, Precision, Recall:

1. Accuracy

- Measures how often the bot correctly categorizes and resolves complaints.
- Depends on **AI model performance**, **database reliability**, and **proper API integration**.
- If the bot fetches incorrect responses, **accuracy decreases**.

2. Precision

- Evaluates how many **retrieved complaints** are actually relevant.
- High precision means the bot **does not misclassify** general queries as complaints.
- Can be improved by **fine-tuning the AI model** and refining **complaint classification algorithms**.

3. Recall

- Measures how many **valid complaints** the system correctly identifies.
- High recall ensures **all genuine complaints** are addressed.
- May suffer if **some complaints are ignored** due to strict filtering.

14. Demo video link

https://drive.google.com/drive/folders/1KhC5-NPE9O_qLj6Rn14-nJHHApPzrauq

15. Individual Contributions

There are **5 members** in our group. Each person had their own role and helped in different parts of the project.

- **Rajesh and Sagar** focused on the coding part. They worked on AI development and report preparation, ensuring the functionality of key components.
- **Nishith and Kartik** handled architecture design using specialized software, studied the Railway Madath platform, and contributed to coding.
- **Vidyasagar** created the PowerPoint presentation, organizing content and designing slides for clarity.

16. Impact of the Solution

i. Who Benefits from This Project?

1. Passengers & Commuters

- Provides a **quick and easy** way to **submit complaints** about railway services.
- Allows users to **track complaint status** and receive updates.
- Ensures **transparency** by classifying complaints and forwarding them to the right department.

- Uses **image and voice-based complaints**, making it more accessible.

2. Railway Authorities & Admins

- Centralized **complaint management** system with **real-time data tracking**.
- Helps in **identifying problem areas** and improving services based on trends.
- Automated **email notifications** ensure timely action on complaints.

3. Railway Security & Maintenance Teams

- Can address **security concerns** like theft, harassment, or staff misbehavior **promptly**.
- Allows better tracking of **cleanliness, water availability, and infrastructure** issues.

4. Government & Regulatory Bodies

- Helps in **policy-making** by analyzing passenger complaints and feedback.
- Ensures **accountability and service improvement** in railway operations.

5. AI & Data Science Enthusiasts

- Demonstrates **real-world AI and automation** using **Streamlit, MySQL, and Google Gemini AI**.
- Uses **image and text analysis** for automated approval/rejection of complaints.
- Integrates **email automation, speech recognition, and database handling** in a single system.

Real-World Impact:

For Passengers

- **Faster Complaint Resolution:** No need to visit railway offices or wait in long queues.
- **Improved Safety & Security:** Quick reporting of theft, harassment, or misconduct.
- **Accessibility:** Voice-based complaints help people who have difficulty typing.
- **Transparency:** Passengers can track their complaints and receive status updates.

For Railway Authorities

- **Better Service Monitoring:** Identifies problem areas like **dirty coaches, non-functional ACs, or water shortages**.
- **Efficient Complaint Handling:** Automatic email forwarding ensures complaints **reach the right department**.
- **Data-Driven Decision Making:** Helps plan **maintenance, security upgrades, and policy changes** based on complaints.
- **Prevention of Corruption:** Detects patterns of **bribery, overcharging, or negligence** among railway staff.

For the Public Sector

- **Scalable Beyond Railways:** Can be adapted for **buses, airports, metro systems, and other public services**.
- **Supports Smart Governance:** AI-powered automation improves **citizen engagement and public service efficiency**.
- **Crime Prevention:** Can help law enforcement track **suspicious activities in railway premises**.

Long-Term Benefits

- **Enhanced Passenger Satisfaction**
- **Reduced Manual Workload for Railway Staff**

- **Proactive Problem Solving Using AI & Data Analytics**
- **A More Reliable and Safer Railway Network**

17. Future Enhancements

1. AI & Machine Learning Integration

- **Smart Complaint Classification:** Enhance Natural Language Processing (NLP) models to categorize complaints more accurately.
- **Predictive Maintenance Alerts:** Use AI to identify frequently reported issues (e.g., faulty ACs, unclean coaches) and suggest preventive measures.
- **Sentiment Analysis:** Implement sentiment analysis to prioritize urgent complaints and provide faster resolutions.

2. Multi-Platform Accessibility

- **WhatsApp & Telegram Bot:** Enable passengers to submit complaints through popular messaging apps.
- **Mobile App Integration:** Develop dedicated Android and iOS apps for streamlined complaint submission and tracking.
- **IVR System:** Introduce a voice-based complaint registration system via a toll-free number.

3. Enhanced Security & Verification

- **OTP & Biometric Authentication:** Implement multi-factor authentication to prevent spam complaints.
- **Blockchain for Transparency:** Store complaint records on a blockchain network to ensure data integrity.
- **AI-Based Fake Complaint Detection:** Use machine learning to detect and filter out fraudulent complaints.

4. Improved User Experience

- **Multilingual Support:** Allow complaints to be filed in multiple regional languages for better accessibility.
- **Live Chat Assistance:** Deploy an AI-powered chatbot for instant support and guidance.

- **Real-Time Complaint Tracking:** Provide live updates on complaint status and actions taken.

5. Advanced Data Analytics & Insights

- **Interactive Dashboards:** Develop dashboards for railway authorities to monitor complaints in real-time.
- **Heatmaps for High-Complaint Zones:** Identify and address problem areas based on location data.
- **Automated Reports:** Generate and send periodic reports to railway officials for performance review and decision-making.

6. Integration with Railway Authorities & Law Enforcement

- **Direct Escalation for Critical Issues:** Automatically forward serious complaints (e.g., theft, harassment) to RPF and local police.
- **Coordination with IRCTC & Railway Helplines:** Ensure seamless communication between multiple grievance redressal platforms.

7. Smart Response & Resolution System

- **Auto-Forwarding to Relevant Authorities:** Assign complaints dynamically based on location and department.
- **Automated Resolution Suggestions:** Use AI to recommend standard solutions for common complaints.
- **Self-Service Portal for Minor Issues:** Allow passengers to resolve minor concerns via guided troubleshooting.

8. Environmental & Social Responsibility

- **Sustainability Feedback Mechanism:** Enable reporting of water wastage, power issues, and excessive plastic use.
- **Awareness & Education Campaigns:** Provide passengers with alerts and tips on railway safety, hygiene, and emergency contacts.

18. Conclusion

The Railway Complaint Bot is a significant step toward modernizing the passenger grievance system, making it more efficient, transparent, and user-friendly. By providing a structured digital platform, it eliminates the inefficiencies of traditional complaint mechanisms, allowing passengers to report issues

quickly and accurately. Its intuitive interface, coupled with technology-driven categorization and tracking, ensures that grievances are addressed in a timely manner.

This initiative not only enhances passenger satisfaction but also fosters accountability within railway authorities. By streamlining complaint resolution, reducing paperwork, and improving response times, the Railway Complaint Bot contributes to a more responsive and passenger-centric railway system. Ultimately, it represents a crucial advancement in public service digitalization, ensuring that every passenger's concern is acknowledged and acted upon effectively.

19. References & Citations

1. OpenAI ChatGPT – Used to assist with code structure, explanation writing, formatting, and overall guidance during the project.
2. the website of the indian railway system helped and been a big part in this project
<https://railmadad.indianrailways.gov.in/madad/final/home.jsp>

