



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Project Kit

Title of the Project

Airport Feedback System

Abstract of the Project

The **Airport Feedback System** is a web-based application built with the **MERN stack (MongoDB, Express.js, React, Node.js)**. This system enables passengers to provide feedback on various airport services such as check-in, security, and lounges. It offers a secure login system, real-time feedback submission, and notifications to passengers. Administrators can use an interactive dashboard to manage feedback, analyze trends, and generate reports to improve airport operations. The application ensures secure data storage and provides a seamless user experience with a modern interface.

Generic keyword:

Feedback System, Passenger Experience, Customer Management, Feedback Reporting, Data Analytics

Specific Technology keywords:

MongoDB, Express.js, React.js, Node.js, JavaScript, REST API, MERN Stack, Database Management

Functional Components of the Project :

User Registration and Authentication:

- Passengers can create accounts, log in, and reset passwords through secure authentication (JWT-based).

Feedback Submission:

- Users submit feedback across various airport services like baggage handling and check-in.

Feedback Categories and Rating:

- Feedback is classified into categories with rating scales for better analysis.

Admin Dashboard:

- Admins manage feedback, view reports, and track trends via interactive charts and tables.

Notifications and Alerts:

- Email or SMS alerts are sent to users upon feedback submission and admin responses.

Analytics and Reporting:

- Provides insights into passenger satisfaction levels through dynamic reports and charts.

Secure Database Management:

- Feedback data is securely stored and managed with MongoDB, following data privacy regulations.

REST API Integration:

- API endpoints handle CRUD operations for feedback and user management.



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Functionality

Users of the System

1. Passengers (Customers):

- Submit feedback, view feedback status, and receive notifications.

2. Administrators (Airport Staff):

- Monitor and manage feedback, analyze trends, and generate reports for operational improvements.

Core Functionalities:

- **User Registration and Authentication:** Secure login system using JWT.
- **Feedback Submission:** Allows users to submit feedback across various services.
- **Admin Dashboard:** Manage user feedback and generate reports for service analysis.
- **Analytics and Reporting:** Provides insights on passenger satisfaction levels.
- **REST API Integration:** CRUD operations using APIs for smooth communication between frontend and backend.
- **Notifications:** Sends alerts for feedback submissions and resolutions.
- **Responsive UI:** Developed with React.js for seamless user experience on multiple devices.

The following steps will be helpful to start off the project –

1. Learn the MERN Stack: Familiarize yourself with MongoDB, Express.js, React.js, and Node.js.
2. Understand the Domain: Speak to airport staff to gain insights into feedback processes.
3. Design User Roles and Workflows: Plan how passengers and admins will use the system.
4. Develop User-Friendly UI: Ensure a clean, responsive design with React.js components.
5. Set Up Notifications and Analytics: Implement email/SMS notifications and feedback trend analysis.

Requirements

Hardware Requirements :

Number	Description	Alternative(if Available)
1.	Processor: x86-64 bit CPU	
2.	RAM: 4GB, Disk Space: 5GB	



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Software Requirements :

Numbers	Descriptions	Alternatives(if Available)
1.	User Interface: Windows OS	Linux, macOS
2.	Development Tools: VS Code, MongoDB, Node.js	
3.	Version Control: Git	GitHub, GitLab

Manpower requirements :

2 to 3 students can complete this in 4 – 6 months if they work fulltime on it.

Milestones and Timelines

Number	Milestone name	Milestone Description	Timeline Week no. From the start of the project	Remarks
1.	Requirements Specification	Document requirements and make a presentation.	2-3	Include any additional functionalities you find relevant.
2.	Technology familiarization	Study MERN stack technologies for practical implementation.	4	Focus on applying the tech to the project rather than theory.
3.	Database creation	Create a MongoDB database with sample user and feedback entries.	5-6	Finalize the schema at this stage.

				proceed with the actual
4.	High-Level Design	High-level and Listing do Detailed possible scenarios Design and then coming with flow-charts or pseudocode to handle the scenario.	7–8	Ensure scenarios align with requirements.
5.	Frontend Development	Build the React.js frontend with login and feedback forms.	9-11	Start drafting a test plan for the system.
6.	API Development	Implement REST APIs with Express.js and Node.js.	12–13	Ensure backend APIs communicate with the frontend correctly.
7.	Integration	The system should be Testing thoroughly tested by running all the testcases written for the system (from mil5). Issues found during	14-15	Another 2 weeks should be there to handle any issues found during testing of the system. After that, the final demo can be arranged.
8.	Final Review	the previous milestone are fixed and the system is ready for the final review.	16-17	During the final review of the project, it should be checked that all the requirements specified during milestone number 1 are fulfilled (or appropriate reasons given for not fulfilling the same)

Guidelines and Reference :

1. MERN Stack Documentation:

<https://mern.io>

2. JavaScript: The Definitive Guide –

David Flanagan

3. MongoDB Documentation:

<https://www.mongodb.com/docs>

4. Express.js Guide:

<https://expressjs.com>

5. React.js Documentation:

<https://reactjs.org>

6. Node.js Documentation:

<https://nodejs.org>