MERN Stack Training
Company: Sensation Software Solutions
Student Name: Tajinder Kaur
Training Duration: 6 Months
Days: 102

Objective of the Day

The objective of **Day 102** was to **implement and polish the Contact Form functionality** while performing **overall backend polishing** for the JourneyJoy Tour & Travel Booking System. The focus was on ensuring that user communication, data validation, backend stability, and API consistency meet **industry-level standards**.

This day was dedicated to refining existing features rather than adding new ones, which is a common practice before project handover or deployment.

Understanding Contact Form Requirements

Before implementation, the Contact Form requirements were reviewed:

- Allow users to send inquiries or messages
- Store contact messages securely in the database
- Validate user inputs
- Provide confirmation feedback to users
- Ensure backend APIs follow consistent response structure

The goal was to make the Contact Form both **functional and professional**.

Contact Form Frontend Enhancements

1. Contact Form UI Review

The Contact page already had a basic UI, which was enhanced:

- Improved input field spacing
- Clear labels for Name, Email, Subject, and Message
- Professional button styling
- Responsive layout using Tailwind CSS

This ensured a clean and approachable interface for users.

---

2. Frontend Validation

Frontend validation logic was added:

- Mandatory field checks
- Email format validation
- Minimum message length validation

Users received immediate feedback, reducing incorrect submissions.

Backend Contact API Implementation

1. Contact Model Design

A dedicated **Contact model** was reviewed and finalized with fields:

- Name
- Email
- Subject
- Message
- Created At timestamp

This structure ensured maintainability and future expansion.

---

2. Contact API Endpoint

A backend API endpoint was implemented to handle contact submissions.

Backend responsibilities:

- Validate request data
- Save contact message to MongoDB
- Return success or error response
- Log errors safely for debugging

This ensured secure data handling.

---

2. Validation & Error Handling Improvements

Backend validation was strengthened for:

- Missing fields
- Invalid data formats
- Server-side errors

Frontend–Backend Synchronization

Special focus was given to:

- Matching frontend payload structure with backend expectations
- Handling backend error responses gracefully
- Displaying confirmation messages to users after contact submission

This improved reliability and reduced runtime issues.

Testing & Hands-on Practice

Manual Testing

Test cases included:

- Valid contact form submission
- Invalid email format

- Empty message submission
- Backend failure simulation

All scenarios were handled gracefully.

User Confirmation Handling

After successful submission:

- User received a confirmation message
- Form fields were reset

This improved user trust and satisfaction.

Project Readiness Impact

With Contact Form and backend polish completed:

- Project stability improved
- Code quality enhanced
- User interaction features finalized

The project moved closer to final deployment readiness.

Conclusion

Day 102 successfully implemented and polished the **Contact Form functionality** while performing essential **backend optimizations**. This day strengthened application reliability, security, and user interaction handling.