

MERN Stack Training

Company: Sensation Software Solutions

Student Name: Tajinder Kaur

Training Duration: 6 Months

Days: 99

Objective of the Day

The objective of Day 99 was to **design, implement, and integrate the Login and Signup authentication system** for the *JourneyJoy Tour & Travel Booking System*. The primary focus was on creating a **secure, user-friendly authentication flow** that allows users to register, log in, and access booking-related features.

Overview of Authentication Work

On Day 99, authentication was treated as a **real-world production feature**, not just a basic form. The following major tasks were covered:

- Signup (User Registration) UI & API
- Login UI & API
- JWT token generation and handling
- Frontend validation and error feedback
- Auth-based UI behavior
- Testing login/signup flows

This day significantly improved the **professional completeness** of the JourneyJoy project.

Signup (User Registration) Implementation

1. Signup UI Design

A dedicated **Signup page** was created with a clean and professional layout.

Signup form fields included:

- Full Name
- Email Address
- Password
- Confirm Password

UI improvements:

- Proper labels and placeholders
- Password visibility toggle (UI level)
- Tailwind-based spacing and alignment
- Responsive design for mobile and desktop

The design ensured that new users could easily understand and complete the registration process.

2. Frontend Validation for Signup

Frontend validation was implemented to improve data accuracy before sending requests to the backend.

Validation rules:

- All fields are mandatory
- Email format validation
- Password minimum length check
- Password and Confirm Password match check

Error messages were shown instantly, improving user experience and reducing backend load.

3. Backend Signup API Integration

On the backend, a **Signup API endpoint** was implemented using **Node.js and Express**.

Backend logic included:

- Checking if the user already exists
- Hashing passwords using bcrypt
- Saving user data securely in MongoDB
- Sending success or error responses

The API returned structured responses, making frontend handling easier.

Login Implementation

1. Login UI Design

A separate **Login page** was created for registered users.

Login form fields:

- Email Address
- Password

UI features:

- Clean and minimal design
- Clear call-to-action button
- “Don’t have an account? Signup” navigation link
- Responsive layout

This ensured a smooth entry point for returning users.

2. Frontend Login Validation

Basic validation was added:

- Empty field check
- Email format verification

Incorrect inputs triggered immediate error messages, guiding users effectively.

3. Backend Login API

The backend **Login API endpoint** was responsible for authentication.

Backend tasks:

- Verifying user email
- Comparing hashed passwords
- Generating JWT token on success
- Sending user data and token to frontend

JWT ensured secure session management and future role-based access control.

Logout Functionality

Logout feature was implemented by:

- Clearing JWT token from storage
- Resetting user state
- Redirecting user to Home page

This ensured session safety and user control.

Error Handling & UX Improvements

Special attention was given to error handling:

- Incorrect password
- User not found
- Duplicate email during signup
- Server error handling

Error messages were displayed clearly without exposing sensitive backend details.

Conclusion

Day 99 was a **major milestone** in the JourneyJoy project. With the successful implementation of **Login and Signup functionality**, the application moved closer to a complete travel booking platform. Authentication laid the foundation for protected features like booking management and user dashboards.