MERN Stack Training
Company: Sensation Software Solutions
Student Name: Gurvinder Singh
Training Duration: 6 Months
Days: 96

Objective of the Day

The objective of **Day 96** was to formally initiate the **JourneyJoy Tour & Travel Booking System project** and begin development with a strong focus on **frontend project setup and basic application structure**. Since this project was assigned during the final phase of training, the primary goal was to quickly establish a clean, scalable frontend foundation that could later support backend integration.

Another important objective of the day was to understand the **functional scope of the JourneyJoy system**, including tour listing, booking flow, authentication, and contact management, and to translate these requirements into a practical frontend layout using modern React development practices.

Project Introduction and Requirement Understanding

At the start of the day, the trainer introduced the **JourneyJoy project** as a real-world inspired **Tour & Travel Booking platform**, similar to commercial travel websites. The system is intended to allow users to explore tour packages, view detailed tour information, submit booking requests, and contact the company for inquiries.

The key features discussed included:

- Home page with featured tours
- Tours listing page
- Tour details page
- Booking form
- User authentication (Login & Signup)
- Contact page
- Admin-side management (planned for backend phase)

The trainer clarified that due to limited time in the final days, the project would be completed in **phases**, starting with frontend-heavy development, followed by partial backend integration.

Frontend Technology Stack Selection

For frontend development, the following technologies were finalized:

- **React.js** – for component-based UI development
- **Vite** – for fast development server and build process
- **Tailwind CSS** – for utility-first styling and responsiveness
- **React Router DOM** – for client-side routing

The trainer explained why Vite is preferred over traditional setups due to its speed and simplicity, especially for rapid project initialization.

Frontend Project Initialization

The JourneyJoy frontend project was initialized using **Vite with React template**. Proper folder naming and project organization were maintained to reflect professional standards.

Initial setup tasks included:

- Creating the project directory for JourneyJoy
- Installing required dependencies
- Running the development server to verify setup
- Cleaning default boilerplate code

This step ensured that the project environment was stable before moving forward.

Tailwind CSS Configuration

After project initialization, **Tailwind CSS** was installed and configured. The following steps were performed:

- Installing Tailwind and PostCSS dependencies
- Configuring `tailwind.config.js`
- Adding Tailwind directives to global CSS
- Verifying Tailwind utility classes in components

The trainer emphasized that Tailwind helps maintain consistent UI styling and speeds up development without writing custom CSS repeatedly.

Basic Component Structure Creation

Once the styling setup was complete, the next focus was on creating the **core frontend components** required across the application.

The following components/pages were created:

- **Header** – containing navigation menu
- **Footer** – containing basic site information
- **Home** – landing page
- **Tours** – tour listing page
- **About** – static informational page
- **Contact** – inquiry form page
- **Login** – user authentication UI
- **Signup** – new user registration UI

These components were created as functional React components following clean naming conventions.

React Router Integration

To enable page navigation, **React Router DOM** was installed and configured. Routes were defined for:

- `/` – Home
- `/tours` – Tours
- `/about` – About
- `/contact` – Contact
- `/login` – Login
- `/signup` – Signup

The routing setup allowed smooth navigation between pages without full page reloads, providing a modern single-page application experience.

Header Navigation Menu Development

The **Header component** was designed to include:

- Website logo/title (JourneyJoy)
- Navigation links: Home, Tours, About, Contact
- Authentication links: Login, Signup

Tailwind CSS classes were used to style the header with proper spacing, alignment, and hover effects. Active navigation behavior was tested to ensure correct routing.

Static Home Page Layout Design

A basic **static Home page layout** was designed to serve as the landing page of the application. The Home page included:

- A hero section with welcome text
- Placeholder area for featured tours
- Call-to-action buttons

Although dynamic data was not yet integrated, placeholders were intentionally added to plan future API-based rendering.

Hands-on Practice and Testing

Throughout the day, continuous testing was performed to ensure stability:

- Verified routing between all pages
- Tested navigation links for correctness
- Checked Tailwind responsiveness on different screen sizes
- Ensured no console errors during navigation

Minor layout adjustments were made to improve alignment and spacing.

Challenges Faced

Some minor challenges were encountered:

- Tailwind configuration path issues initially
- Adjusting responsive behavior of navigation menu
- Structuring components for future scalability

These were resolved through trainer guidance and debugging.

Learning Outcome

By the end of Day 96, I gained:

- Clear understanding of JourneyJoy project requirements
- Hands-on experience with rapid React + Vite setup
- Practical knowledge of Tailwind CSS integration

- Experience in structuring a multi-page frontend application

---

Conclusion

Day 96 marked the **official beginning of the JourneyJoy Tour & Travel Booking System project**. The day focused entirely on **frontend foundation setup**, including project initialization, component structure creation, routing configuration, and basic UI layout. This strong foundation ensured that upcoming days could focus on UI enhancement and backend integration without structural issues. Daily diary documentation was maintained as required, even though backend work was planned for later stages.