

Training Day 21

Day 21- 17th July 2025

Topics Covered: Backend and Frontend – APIs and Tools

1. Making Protected Routes (Header-Based Authentication)

- **What is a protected route?**
- A route that requires the user to be authenticated before accessing it.
- **How to make it protected?**
 - User logs in and receives a JWT token.
 - Client sends requests to protected routes with token in Authorization header:

Authorization: Bearer <token>

- In backend, create middleware to check token:
 - Extract token from header.
 - Verify it using JWT.
 - If valid → allow access to route.
 - If invalid → return an error.
- **Why use protected routes?**
 - Keeps sensitive data or user-specific actions secure.
 - Ensures only authorized users can access protected endpoints.

2. Toastr

- **What is Toastr?**
 - A JavaScript library for non-blocking notifications (pop-up alerts) in web apps.
- **Why use Toastr?**
 - Provides quick success, error, or info messages.
 - Attractive, user-friendly styling without blocking the UI.

3. Axios

- **What is Axios?**
 - A popular HTTP client library for making API requests from frontend (React, Angular, etc.) or Node.js.
- **Why use Axios?**
 - Simple syntax for **GET, POST, PUT, DELETE** requests.
 - Returns **Promises** for easy asynchronous handling.
 - Automatically converts data to/from **JSON**.

4. CORS (Cross-Origin Resource Sharing)

- **What is CORS?**
 - A browser security feature that restricts requests from one origin to another unless explicitly allowed by the server.
- **Why do we need it?**
 - Example: Frontend on localhost:3000 requesting backend API on localhost:5000.
 - Without CORS enabled, the request will be blocked.
- **How to handle it in Express.js?**

```
const cors = require('cors');  
app.use(cors());
```

Tools Used

- **Visual Studio Code (VS Code)**
- **Node.js, Express.js**
- **React (Frontend – using Axios & Toastr)**
- **JWT for Authentication**
- **CORS package**