

|  |  |
| --- | --- |
| **Month 1: Node and Express Basics** | |
|  | |
| **Week 1 : Foundations of Node.js** | |
| Day 1 | Introduction to Node.js |
| Day 2 | Node.js Basics |
| Day 3 | Asynchronous Programming |
| Project | Create a command-line tool that allows users to perform basic file management operations |
|  | |
| **Week 2 : Core Node.js Concepts** | |
| Day 4 | File System Operations |
| Day 5 | Networking and HTTP Basics |
| Day 6 | Advanced Node.js Concepts |
| Project | Develop a basic node.js web server that responds with "Hello, Express!" on visiting the homepage ("/") |
|  | |
| **Week 3 : Introduction to Express** | |
| Day 7 | Overview of Express framework |
| Day 8 | Setting up a basic Express project |
| Day 9 | Handling routes and HTTP methods |
| Project | Create a simple Express.js web server that responds with "Hello, Express!" on visiting the homepage ("/") |
|  | |
| **Week 4 : Middleware and Templating** | |
| Day 10 | Using EJS as templating engines |
| Day 11 | Understanding middleware in Express |
| Day 12 | Implementing middleware for authentication and error handling |
| Project | Implement a middleware that logs each request method and URL to the console |

|  |  |
| --- | --- |
| **Month 2: Advanced Concepts** | |
|  | |
| **Week 5 : Working with Data** | |
| Day 13 | Introduction to databases and PostgreSQL |
| Day 14 | Setting up PostgreSQL database and connecting to Express |
| Day 15 | Creating, reading, updating, and deleting data (CRUD operations) |
| Project | Build a basic application where users can view items (like tasks or notes) stored in a PostgreSQL database |
|  | |
| **Week 6 : Advanced Database Operations** | |
| Day 16 | Querying data with PostgreSQL |
| Day 17 | Implementing joins and transactions |
| Day 18 | Using Sequelize ORM for database interactions |
| Project | Extend the application to include advanced features like filtering for database interactions |
|  |  |
| **Week 7 : Authentication and Authorization** | |
| Day 19 | Managing sessions and cookies |
| Day 20 | Implementing user authentication with Passport.js |
| Day 21 | Role-based access control (RBAC) and authorization |
| Project | Implement user authentication using Passport.js. Create a login page and restrict access to certain routes based on user roles (e.g., admin vs. regular user) |
|  |  |
| **Week 8 : Deployment and Performance Optimization** | |
| Day 22 | Integrating WebSocket for real-time communication |
| Day 23 | Performance optimization techniques (caching, compression) |
| Day 24 | Deploying Express applications to platforms like Heroku |
| Project | Add real-time communication using WebSocket to create a chat apps |