🗓 □ 30-Day Backend Developer Plan (Node.js + Express)

□ WEEK 1: Backend & Node.js Basics (Days 1–7)

Day 1

- What is backend?
- Client–server architecture
- HTTP & HTTPS
- Install Node.js & npm

✅ Task: Check `node -v`, `npm -v`

Day 2

- JavaScript for backend
- Callbacks
- Promises
- `async / await`

✅ Task: Write async JS examples

Day 3

- Node.js basics
- NPM
- Modules
- `require()` / `module.exports`

✅ Task: Create simple Node app

Day 4

- File system (`fs`)
- Read/write files
- Environment variables

✅ Task: Read & write a file

Day 5

- Express.js introduction
- Create server
- Routes
- Request & response

✅ Task: Hello World Express app

Day 6

- REST API basics
- HTTP methods
- Status codes

✅ Task: Create GET & POST routes

Day 7

- Middleware
- `express.json()`
- Error handling

✅ Task: Custom middleware

□ WEEK 2: Express + MongoDB (Days 8–14)

Day 8

- What is database?
- SQL vs NoSQL
- MongoDB basics

Day 9

- MongoDB installation
- MongoDB Compass
- CRUD operations

✅ Task: Insert, find, delete data

Day 10

- Mongoose
- Schema

- Model

✅ Task: Create User model

Day 11

- Connect Express with MongoDB
- CRUD APIs

✅ Task: User CRUD API

Day 12

- Folder structure (MVC)
- Controllers
- Routes

Day 13

- Validation
- Error handling
- try–catch

Day 14

- Mini Project 1:
  - **To-Do Backend API**

□ WEEK 3: Authentication & Security (Days 15–21)

Day 15

- Authentication vs Authorization
- Login / Signup flow

Day 16

- Password hashing (bcrypt)

Day 17

- JWT (JSON Web Token)
- Token creation & verification

Day 18

- Protected routes
- Role-based access

Day 19

- Security basics:
  - CORS
  - dotenv
  - Helmet

Day 20

- Postman
- API testing
- Status codes

Day 21

- Mini Project 2:
  - **Auth System (Login / Register API)**

□ WEEK 4: Advanced + Deployment (Days 22–30)

Day 22

- API best practices
- Pagination
- Sorting
- Filtering

Day 23

- File upload (multer)
- Image upload

Day 24

- Cloud storage (Cloudinary)
- Email sending (nodemailer)

Day 25

- Git & GitHub
- Push backend project

Day 26

- MongoDB Atlas
- Environment config

Day 27

- Backend deployment
  - Render / Railway

Day 28

- Final Project:
  - Blog / E-commerce Backend

Day 29

- Improve project
- Clean code
- README file

Day 30

- Revise everything
- Mock interview questions
- Plan next step (Full Stack)

□ FINAL PROJECT IDEAS

✓ Blog Backend
✓ User Auth API
✓ E-commerce Backend
✓ Notes App API

🎯 After 30 Days You Can:

✅ Build REST APIs
✅ Handle authentication
✅ Connect database
✅ Deploy backend
✅ Crack backend interviews (junior)

□ Backend Developer Roadmap (Node.js + Express)

□ PHASE 1: Backend Fundamentals (FOUNDATION)

1□ Internet & HTTP Basics

Learn:

- Client–Server Architecture
- HTTP vs HTTPS
- Request & Response cycle
- HTTP Methods
  - GET, POST, PUT, DELETE
- Status Codes
  - 200, 201, 400, 401, 404, 500

☞ **Why important?**
Every backend works on HTTP.

2□ JavaScript for Backend

Focus on:

- Callbacks
- Promises
- `async / await`
- Event loop
- Error handling (`try-catch`)

□ PHASE 2: Node.js Core

3□ Node.js Basics

Learn:

- What is Node.js
- NPM (Node Package Manager)
- `require()` & `module.exports`
- File system (`fs`)
- Environment variables (`process.env`)

4️⃣ Express.js Framework

Learn:

- Creating server
- Routing
- Request & Response
- Middleware
- Error handling
- REST API structure

Example:

```
app.get("/users", (req, res) => {
  res.json({ message: "Users fetched" });
});
```

□ PHASE 3: Database (VERY IMPORTANT)

5️⃣ Database Basics

Understand:

- What is a database
- SQL vs NoSQL
- Collections & Documents

6️⃣ MongoDB + Mongoose

Learn:

- MongoDB CRUD operations
- Schema & Model
- Relationships
- Validation

Example:

```
const User = mongoose.model("User", userSchema);
```

□ PHASE 4: Authentication & Security 🔐

7️⃣ Authentication

Learn:

- Login / Signup flow
- Password hashing (bcrypt)
- JWT (JSON Web Token)
- Authorization (roles)

8️⃣ Security Basics

Learn:

- Environment variables
- CORS
- Rate limiting
- Input validation
- Helmet

□ PHASE 5: Advanced Backend Concepts ⚙️

9️⃣ API Best Practices

Learn:

- MVC architecture
- Pagination
- Filtering & sorting
- API versioning

🔟 File Uploads & Emails

Learn:

- File upload (multer)
- Cloud storage (Cloudinary)
- Sending emails (nodemailer)

□ PHASE 6: Testing & Debugging □

1️⃣1️⃣ Testing

Learn:

- Unit testing
- API testing
- Postman
- Jest (basic)

□ PHASE 7: Deployment 🚀

1️⃣2️⃣ Deployment

Learn:

- Environment setup
- Hosting backend on:
  - Render
  - Railway
- MongoDB Atlas
- Production vs Development

□ PHASE 8: Git & Professional Skills 💼

1️⃣3️⃣ Git & GitHub

Learn:

- git init
- commit
- push
- pull
- branches

1️⃣4️⃣ System Design (Beginner)

Learn:

- REST API design
- Monolithic vs Microservices
- Load balancing (basic idea)

□ Backend Project Roadmap (MUST BUILD)

Beginner Projects:

- Simple CRUD API
- To-Do Backend

Intermediate Projects:

- User Authentication API
- Blog Backend
- E-commerce Backend

Advanced:

- Role-based access
- Payment integration

□ FINAL ROADMAP SUMMARY

```
JavaScript
   ↓
Node.js
   ↓
Express
   ↓
MongoDB
   ↓
Authentication
   ↓
Security
   ↓
Projects
   ↓
Deployment
```

🎯 What to Do NEXT (Right Now)

✅ Install Node.js
✅ Build first Express server
✅ Learn REST APIs
✅ Push code to GitHub
