**🚀 Upgraded 14-Day JavaScript + Full Stack Bootcamp**

**Designed for aspiring full stack devs with strong fundamentals**

**🎯 Final Project: Game Feedback Management System**

* Frontend: Customizable feedback form for players
* Backend: Admin dashboard to manage feedback
* Stack: Vanilla JS, Node.js, Express, MongoDB, optional React rewrite

**🟩 Phase 1: Core JS + DOM + Frontend (Days 1–4 ✅ Completed)**

**📅 Day 1: JavaScript Essentials I – The Foundation**

* Variables (let, const, scope)
* Primitive types
* Functions (declaration, arrow)
* Conditionals, loops
* Arrays and objects

**Challenge:** Utility functions: removeDuplicates, flatten, deepClone

**📅 Day 2: JavaScript Essentials II – Data Mastery**

* Destructuring, spread/rest
* Array methods: map, filter, reduce
* String methods
* Closures, HOFs

**Challenge:** groupBy, getNested

**📅 Day 3: DOM & Events – The Interactive Browser**

* DOM navigation, event listeners, bubbling
* Form inputs, buttons, data handling
* localStorage basics

**Challenge:** Comment board app with add/edit/delete + localStorage

**📅 Day 4: Async JavaScript + Fetch**

* Callbacks, Promises, async/await
* Fetch API, try/catch, error handling

**Challenge:** Pokémon viewer using pokeapi.co with images/stats/error handling

**🟨 Phase 2: Backend & Fullstack Integration (Starting Now)**

**📅 Day 5: Project Kickoff – Structure & Setup (🔁 Now Doing)**

* Initialize backend project
  + npm init -y
  + Install Express, nodemon
  + Setup server.js or index.js
* Folder structure: routes/, controllers/, public/, views/, utils/
* Serve static files (form HTML + JS)
* CORS basics

**Challenge:**

* Create GET / to serve the feedback form
* Create GET /admin to serve a placeholder dashboard (basic HTML or h1)
* Add nodemon script, use express.static() for frontend

**Stretch:**

* Add custom middleware logger for request method + URL
* Discuss separation of concerns (router/controller split)

**📅 Day 6: Express – Routing & Middleware**

* REST routes: GET, POST, DELETE
* express.json() for body parsing
* Return JSON responses
* Use Postman to test routes

**Challenge:**

* /api/reports with in-memory storage
  + GET: return all reports
  + POST: add report to memory
  + (Stretch) DELETE: by ID

**Stretch:**

* Use uuid to assign IDs
* Separate routes/controller logic

**📅 Day 7: MongoDB – Persistent Storage**

* Intro to MongoDB and Mongoose
* Connecting to MongoDB Atlas
* Creating Report model
* Replacing memory with DB storage
* Querying by tag/status

**Stretch:** Use .lean() on queries to optimize

**📅 Day 8: Frontend ↔ Backend Wiring**

* fetch() to backend from frontend
* Form submission + JSON POST
* Show dynamic messages, success/error
* Load reports on page load

**Stretch:** Show loading spinner during requests

**📅 Day 9: File Uploads – Screenshots**

* Multer for file handling
* File preview + input in frontend
* Save file path in DB
* Serve static uploaded files

**Security Notes:** Validate file type + size

**📅 Day 10: Admin Dashboard Features**

* Toggle report status
* Filtering, search
* Query params in URL
* Sorting by date

**Stretch:** Pagination using query params

**📅 Day 11: Authentication – Securing Admin**

* JWT-based login
* Login form + token storage
* Protect admin routes via middleware

**Stretch:** Token expiry and refresh logic

**📅 Day 12: Testing & Validation**

* Unit testing with Jest
* API testing with Supertest
* Validation using express-validator

**Stretch:** Postman collection with tests

**📅 Day 13: Deployment & Polish**

* Deploy backend (Render/Railway)
* Deploy frontend (Netlify/Vercel)
* .env config, base URLs, HTTPS

**Stretch:** GitHub Actions CI setup

**📅 Day 14: Documentation & Devlog**

* Write a README with live demo, screenshots, setup steps
* Optional: Record 90s walkthrough video
* Start writing devlog post

**Bonus:** Diagram frontend/backend/data flow for future interviews

**🚀 Optional Week 3+: React Rewrite**

* Convert frontend to React
* Hooks, context, routing
* Reuse backend API