30-Day Brutal Full Stack Project Plan

Rules (No Mercy Edition)

1. **Daily commits required** – even small changes count. No "placeholder" commits.

2. **No copying entire projects** – small snippets for syntax are allowed.

3. **Deploy or die** – every project must be live (Netlify, Vercel, Render, or Heroku).

4. **Log everything** – errors, fixes, learning points. Your GitHub README should reflect actual progress.

Week 1: Foundations in Action

- **Goal:** Stop watching, start coding.

- **Daily tasks:**

  - Day 1: Set up GitHub repo, basic HTML/CSS template. Commit immediately.

  - Day 2: Create a simple **static website** (portfolio/homepage). Deploy live.

  - Day 3: Add a contact form (no backend yet, just HTML + CSS).

  - Day 4: Add interactive features with **vanilla JS** (button click counters, simple input validations).

  - Day 5: Refactor your code: organize folders, clean HTML/CSS. Commit every tiny change.

  - Day 6: Make your first **responsive page** (media queries). Test on phone & PC.

  - Day 7: **Mini-review & deploy**: your first live, functional, responsive site.

Week 2: Backend Basics

- **Goal:** Make your projects dynamic.

- **Daily tasks:**

  - Day 8: Learn **Node.js basics**: set up server, routes, and package.json.

  - Day 9: Connect your Node.js server to a simple **JSON data file** (simulate a database).

- Day 10: Build a small **To-Do App backend**: add, remove, list tasks.

- Day 11: Implement simple **form submission** (contact form now sends data to backend).

- Day 12: Learn basic **Express routing**; integrate with your To-Do app.

- Day 13: Add **error handling** (invalid input, empty fields).

- Day 14: **Deploy backend** (Render/Heroku) and connect your frontend.

Week 3: Database + Authentication

- **Goal:** Real-world features.

- **Daily tasks:**

  - Day 15: Learn **MongoDB** basics (install, connect, CRUD).

  - Day 16: Connect your To-Do app to MongoDB (store tasks in DB instead of JSON).

  - Day 17: Add **user registration & login** (hash passwords with bcrypt).

  - Day 18: Add **session management or JWT authentication**.

  - Day 19: Protect routes so only logged-in users can add/view tasks.

  - Day 20: Add **input validation + error messages** in backend.

  - Day 21: **Deploy full backend + database**. Make sure everything works live.

Week 4: Polishing & Extra Features

- **Goal:** Make it professional, impressive, deployable.

- **Daily tasks:**

  - Day 22: Add **frontend integration** with backend API (AJAX/fetch).

  - Day 23: Make the app **responsive & visually polished** (CSS frameworks like Tailwind or Bootstrap optional).

  - Day 24: Add a **search/filter feature** in To-Do app.

  - Day 25: Add **user-friendly notifications** (success/failure alerts).

- Day 26: Write a **README.md** for the project with screenshots, features, and instructions.

- Day 27: Add **GitHub issues & milestones** for features you haven't finished.

- Day 28: **Final bug fixes & polish**: mobile check, load times, styling.

- Day 29: Share your project with friends/Reddit/Discord; get feedback.

- Day 30: **Deploy & celebrate**: everything live, GitHub updated, your soul slightly less broken.

Outcome After 30 Days

- ✅ 1 fully functional **full stack project** (frontend + backend + database + authentication).

- ✅ Live deployment with proper domain or link.

- ✅ GitHub repo with actual commits, documentation, and issues.

- ✅ Your portfolio now screams: "I can actually build, not just watch tutorials."