

MERN Developer Job-Ready Checklist

Phase 0 — Setup & Mindset (1-2 days)

- Install VS Code, Node.js, Git, Chrome
- Learn folder structure and terminal basics
- Install Git, setup GitHub account
- Learn basic Git commands:
 - `git init`
 - `git add .`
 - `git commit -m "initial commit"`
 - `git push origin main`
- Learn how to run Node server & React dev server **Outcome:** Dev environment ready, can run server + track code with Git

Phase 1 — JavaScript Core (2-3 weeks)

- Variables, loops, functions, arrays, objects
- DOM manipulation & events
- ES6+ features (let/const, template literals, destructuring, spread/rest)
- Async JS: fetch, Promises, async/await **Exercises:**
- To-do list app (vanilla JS)
- Fetch JSON & display on page
- Mini calculator or counter **Git/GitHub:**
- Push every exercise/project to GitHub
- Commit messages: clear & descriptive
- Use branches for experiments

Phase 2 — HTML + CSS (2 weeks)

- HTML: semantic tags, forms, tables
- CSS: box model, flexbox, grid, responsive design
- Event handling with JS **Exercises:**
- Portfolio page
- Responsive landing page
- Form validation **GitHub:**
- Push projects as separate repos
- Add README.md describing the project

Phase 3 — React Fundamentals (2-3 weeks)

- Components, JSX, props, state
- Conditional rendering, lists, forms, API calls (fetch/axios) **Exercises:**
- Convert vanilla JS projects to React
- Build small dashboard apps **GitHub/LinkedIn:**
- Update GitHub with React repos

- LinkedIn Post Template: "Built a small React project using components, state & API calls. Excited to keep improving my frontend skills! #React #WebDevelopment"

Phase 4 — Node + Express Backend (2–3 weeks)

- Routes (GET, POST, PUT, DELETE)
- Middleware, JSON handling, error handling
- Environment variables (dotenv) **Exercises:**
- CRUD API for users or tasks
- Connect backend to React frontend **GitHub:**
- Push backend repos
- Use .gitignore properly
- Write clear commit messages **LinkedIn:**
- "Built my first backend API with Express + Node, able to do CRUD operations. #NodeJS #Express #FullStack"

Phase 5 — Database (MongoDB) (2 weeks)

- Collections, Documents, Queries
- Mongoose ORM: Models & Schemas
- Connect backend to MongoDB **Exercises:**
- User management API with MongoDB
- Integrate with React frontend **GitHub:**
- Push full-stack MERN repo
- Add instructions in README.md for running the project **LinkedIn:**
- "Completed a full-stack MERN project integrating MongoDB with React & Node. #MERN #FullStackDeveloper"

Phase 6 — Full MERN Projects (3–4 weeks)

- Real-world projects:
- Student Management System
- Donation/Community Platform (Eclat Unites project)
- Booking/Tracking App
- Skills: Frontend + backend integration, API design, state management, error handling, project structure **GitHub:**
- Each project = one repo
- Add README.md: overview, features, technologies, how to run **LinkedIn:**
- Post project update: Screenshots / GIF demo + 2–3 lines description + invite feedback

Phase 7 — GitHub Mastery + LinkedIn Branding (Ongoing)

GitHub: - Use branches for experiments - Pull requests (even for your own projects) - Add Issues & TODO for better project management - Tag releases for project versions **LinkedIn:** - Weekly updates on learning progress - Share small demos, challenges solved, insights learned - Connect with other devs, comment on tech posts - Profile optimization: - Headline: "Aspiring Full-Stack Developer | MERN | React + Node + MongoDB" - Experience: Projects, Roles (self-learning, freelance, or personal projects)

Phase 8 — Deployment & Professional Touch (1–2 weeks)

- Deploy frontend (Vercel) and backend (Render/Heroku)
- Add live demo links in GitHub repos
- Optimize projects: responsiveness, error handling, UI
- Add README badges (Node version, MongoDB, React) **LinkedIn:**
- Post live demo link of each project
- Post a “Portfolio complete” post with screenshots, tech stack, and GitHub/Live links

Notes: - Follow exercises and push everything to GitHub. - Use LinkedIn for **weekly highlights** instead of daily posts. - Keep commit messages and repos clean; this becomes your portfolio. - Focus on understanding fundamentals before moving to complex projects.