A **complete, practical 6-month day-by-day roadmap for Full Stack Development** — broken down into **Daily Tasks** with clear topics and milestones.

**📅 6-Month Full Stack Developer Roadmap**

**📌 🗓️ Month 1 — HTML, CSS, and Basic JavaScript**

**Goal:** Build a strong foundation in web basics.

**Week 1: HTML**

* **Day 1:** Internet basics, how the web works, client-server model.
* **Day 2:** HTML structure, tags, elements, attributes.
* **Day 3:** Lists, tables, images.
* **Day 4:** Links, forms, input types.
* **Day 5:** Semantic HTML: header, nav, article, section.
* **Day 6:** Media elements: video, audio, iframes.
* **Day 7:** Mini project — Build a personal resume website.

**Week 2: CSS Basics**

* **Day 8:** CSS syntax, selectors, colors.
* **Day 9:** Box model, borders, margins, padding.
* **Day 10:** Positioning, floats, display.
* **Day 11:** Flexbox basics.
* **Day 12:** Grid layout basics.
* **Day 13:** Fonts, icons, Google Fonts.
* **Day 14:** Mini project — Style your resume website.

**Week 3: Advanced CSS**

* **Day 15:** Responsive design, media queries.
* **Day 16:** CSS variables.
* **Day 17:** CSS animations, transitions.
* **Day 18:** Pseudo-elements and pseudo-classes.
* **Day 19:** CSS frameworks: Bootstrap intro.
* **Day 20:** Build with Bootstrap — Navbar, Cards.
* **Day 21:** Mini project — Responsive portfolio site.

**Week 4: JavaScript Basics**

* **Day 22:** JS syntax, variables, let, const.
* **Day 23:** Data types, operators.
* **Day 24:** Conditionals, loops.
* **Day 25:** Functions and scope.
* **Day 26:** Arrays, objects.
* **Day 27:** DOM manipulation basics.
* **Day 28:** Mini project — To-do list.

**📌 🗓️ Month 2 — JavaScript Deep Dive & Version Control**

**Week 5: Advanced JS**

* **Day 29:** ES6 — Arrow functions, template literals.
* **Day 30:** Destructuring, spread/rest operators.
* **Day 31:** Callbacks, Promises.
* **Day 32:** Async/Await.
* **Day 33:** JSON, localStorage.
* **Day 34:** Fetch API, AJAX basics.
* **Day 35:** Mini project — Weather app with API.

**Week 6: Git & GitHub**

* **Day 36:** Version control basics.
* **Day 37:** Git init, add, commit.
* **Day 38:** Branching and merging.
* **Day 39:** Pushing to GitHub.
* **Day 40:** Pull requests, issues.
* **Day 41:** Clone, fork, collaboration.
* **Day 42:** Mini project — Host your projects on GitHub.

**Week 7: Package Managers & Tooling**

* **Day 43:** NPM, installing packages.
* **Day 44:** Scripts, dependencies.
* **Day 45:** Intro to build tools — Webpack basics.
* **Day 46:** Babel basics.
* **Day 47:** ESLint, Prettier.
* **Day 48:** VSCode extensions & productivity.
* **Day 49:** Review & mini project cleanup.

**Week 8: Advanced JS Project**

* **Day 50–56:** Build a complete JS project:
  + Weather app or Movie search (API, fetch, localStorage, styling)

**📌 🗓️ Month 3 — Frontend Framework (React)**

**Week 9: React Basics**

* **Day 57:** What is React? JSX, create-react-app.
* **Day 58:** Components, props.
* **Day 59:** State, events.
* **Day 60:** Conditional rendering.
* **Day 61:** Lists and keys.
* **Day 62:** Forms in React.
* **Day 63:** Mini project — Counter, simple form.

**Week 10: React Deep Dive**

* **Day 64:** useEffect hook.
* **Day 65:** Context API.
* **Day 66:** Routing with React Router.
* **Day 67:** Lifting state up.
* **Day 68:** Reusable components.
* **Day 69:** Styling in React — CSS Modules, Styled Components.
* **Day 70:** Mini project — Multi-page app with routing.

**Week 11: API & State Management**

* **Day 71:** Fetching API data in React.
* **Day 72:** Axios vs Fetch.
* **Day 73:** Redux intro.
* **Day 74:** Redux toolkit basics.
* **Day 75:** Connecting React with Redux.
* **Day 76:** Redux DevTools.
* **Day 77:** Mini project — Blog or Todo with Redux.

**Week 12: React Project**

* **Day 78–84:** Build a complete React project:
  + E-commerce product page, movie search, or blog.

**📌 🗓️ Month 4 — Backend (Node.js + Express)**

**Week 13: Node.js Basics**

* **Day 85:** What is Node.js? Setup, REPL.
* **Day 86:** Modules, exports, require.
* **Day 87:** File system.
* **Day 88:** Events and streams.
* **Day 89:** NPM in Node.js.
* **Day 90:** Build a simple CLI app.
* **Day 91:** Mini project — File manipulator CLI.

**Week 14: Express.js**

* **Day 92:** Intro to Express.
* **Day 93:** Routes and middleware.
* **Day 94:** Request & response objects.
* **Day 95:** Serving static files.
* **Day 96:** Templating engines.
* **Day 97:** REST API basics.
* **Day 98:** Mini project — CRUD API.

**Week 15: Databases (MongoDB)**

* **Day 99:** Intro to NoSQL, MongoDB.
* **Day 100:** Install MongoDB locally & Atlas.
* **Day 101:** CRUD operations.
* **Day 102:** Mongoose intro.
* **Day 103:** Models & schemas.
* **Day 104:** Connecting Express to MongoDB.
* **Day 105:** Mini project — Full CRUD with Express & MongoDB.

**Week 16: Authentication**

* **Day 106:** Intro to Auth — JWT & Cookies.
* **Day 107:** User signup/login.
* **Day 108:** Protecting routes.
* **Day 109:** Password hashing with bcrypt.
* **Day 110:** OAuth basics.
* **Day 111:** Testing API with Postman.
* **Day 112:** Mini project — Auth API.

**📌 🗓️ Month 5 — Full Stack Integration**

**Week 17: Connecting Frontend & Backend**

* **Day 113:** CORS, Proxy, API routes.
* **Day 114:** Connect React to Express.
* **Day 115:** Handle requests & responses.
* **Day 116:** Env variables.
* **Day 117:** Error handling.
* **Day 118:** Deployment basics.
* **Day 119:** Mini project — MERN Stack CRUD.

**Week 18: Advanced Full Stack Features**

* **Day 120:** File uploads.
* **Day 121:** Image hosting.
* **Day 122:** Realtime with Socket.IO.
* **Day 123:** Payment gateways.
* **Day 124:** Rate limiting, security.
* **Day 125:** Testing with Jest.
* **Day 126:** Mini project — Chat app.

**Week 19–20: Full Stack Capstone Project**

* **Day 127–140:** Build a real full stack app.
  + Example: Social media, e-commerce, or booking system.
  + Include auth, CRUD, payments, file uploads.

**📌 🗓️ Month 6 — DevOps, Deployment, & Interview Prep**

**Week 21: Deployment**

* **Day 141:** Hosting frontend (Vercel, Netlify).
* **Day 142:** Hosting backend (Render, Railway).
* **Day 143:** MongoDB Atlas, Cloudinary.
* **Day 144:** Domain & DNS.
* **Day 145:** HTTPS, SSL basics.
* **Day 146:** CI/CD pipelines.
* **Day 147:** Deploy your capstone app.

**Week 22: Testing & Best Practices**

* **Day 148:** Unit testing basics.
* **Day 149:** Integration testing.
* **Day 150:** Test React with Jest.
* **Day 151:** Test API with Postman & Supertest.
* **Day 152:** Code linting & formatting.
* **Day 153:** Performance optimization.
* **Day 154:** Project audit.

**Week 23–24: Interview Prep**

* **Day 155–168:**
  + Revise JS & React.
  + DSA basics (arrays, strings, objects).
  + Practice 1–2 LeetCode daily.
  + Mock interviews.
  + Prepare portfolio & resume.
  + Apply for internships/junior roles!

**✅ By the end of 6 months, you will have:**

* Solid understanding of **Frontend (HTML, CSS, JS, React)**.
* Backend knowledge with **Node.js + Express**.
* Database experience with **MongoDB**.
* A **deployed full stack project**.
* **Git/GitHub** proficiency.
* Interview preparation done!

If you’d like, I can convert this into a **Google Sheet**, **PDF**, or a **Notion Template** — want me to prepare it for you? 🚀✨