**Full Stack Developer Roadmap**

**Phase 1: Web Fundamentals**

1. **How the Internet Works**
   * HTTP/HTTPS protocols, client-server model
   * DNS, IPs, hosting, CDNs
2. **Browser Mechanics**
   * Rendering engine, DOM, CSSOM, JavaScript execution
   * DevTools for debugging and optimization
3. **Basic Tools**
   * Command line basics (bash, shell)
   * Git & GitHub (branches, PRs, commits)

**Phase 2: Frontend Development**

**1. HTML, CSS, JavaScript**

* Semantic HTML5 tags
* CSS3 (Flexbox, Grid, Responsive Design)
* JavaScript: DOM, ES6+, events, fetch API, async/await

**2. Advanced Styling**

* Preprocessors: Sass, Less
* Frameworks: TailwindCSS, Bootstrap
* CSS-in-JS (Styled Components, Emotion)

**3. Modern JavaScript Frameworks**

* **React.js** (most popular)
  + Hooks, State, Context API, React Router
* Alternative: Vue.js, Angular, Svelte

**4. Frontend Tooling**

* Linters (ESLint), Formatters (Prettier)
* Package managers (npm, yarn, pnpm)
* Module bundlers: Vite, Webpack, Parcel

**Phase 3: Backend Development**

**1. Programming Language**

* **JavaScript (Node.js)** OR **Python**, **Go**, **Java**

**2. Node.js & Frameworks**

* Express.js, Nest.js (for advanced apps)
* Handling routes, middleware, cookies, file uploads

**3. RESTful APIs & WebSockets**

* Build CRUD APIs
* Learn status codes, headers, request/response
* Real-time with Socket.IO or WebSockets

**4. Authentication & Security**

* JWT, OAuth2, Sessions
* Bcrypt, Helmet.js, CORS, CSRF protection

**Phase 4: Databases**

**1. Relational Databases**

* PostgreSQL or MySQL
* Schema design, Normalization, Indexing, Joins

**2. NoSQL Databases**

* MongoDB, Redis, Firebase
* Use-cases: unstructured data, caching, real-time data

**3. ORM/ODM Tools**

* Prisma (SQL), Mongoose (MongoDB), TypeORM, Sequelize

**Phase 5: Full Stack Integration**

* Connect frontend to backend using APIs
* Handle API calls (GET, POST, PUT, DELETE)
* Manage state with Context API or Redux
* Error handling, loading states, and authentication flows

**Phase 6: DevOps & Deployment**

1. **Basic DevOps Concepts**
   * CI/CD pipelines (GitHub Actions, GitLab CI)
   * Containers with Docker
   * Environment Variables and Secrets
2. **Hosting & Deployment**
   * Frontend: Vercel, Netlify, GitHub Pages
   * Backend: Railway, Render, Heroku, AWS EC2
   * Databases: MongoDB Atlas, Supabase, PlanetScale

**Phase 7: Testing**

* Unit testing: Jest (JavaScript), Mocha
* Integration testing: Supertest
* E2E testing: Cypress, Playwright

**Phase 8: Advanced Concepts**

1. **TypeScript**
   * Static typing for JS projects, Interfaces, Generics
2. **GraphQL**
   * Querying with Apollo Server & Client
   * Schema design and resolvers
3. **Web Performance Optimization**
   * Lazy loading, bundling, caching, Lighthouse
4. **WebSockets & Real-time**
   * Notifications, chats, real-time dashboards

**Phase 9: System Design & Architecture**

* Monolith vs Microservices
* REST vs GraphQL
* Caching, Load Balancing, Rate Limiting
* Queues: BullMQ, RabbitMQ
* Design patterns (MVC, Factory, Singleton)

**Phase 10: Portfolio & Real Projects**

* **Ideas to build**:
  + Authentication system
  + Blog with CMS
  + Chat app (Socket.IO)
  + E-commerce platform with payment gateway
  + Dashboard with charts and filtering
* **Documentation & Hosting**
  + Use Markdown/Notion for documentation
  + Host portfolio on GitHub and deploy projects

**🔧 Bonus: Tools Checklist**

| **Category** | **Tools** |
| --- | --- |
| Frontend | HTML, CSS, JavaScript, React, Tailwind |
| Backend | Node.js, Express.js, Nest.js |
| Database | PostgreSQL, MongoDB, Redis |
| Deployment | Docker, Vercel, Netlify, Heroku |
| Version Control | Git, GitHub |
| CI/CD | GitHub Actions, Docker Compose |
| Testing | Jest, Cypress, Playwright |
| Authentication | JWT, OAuth, Firebase Auth |
| Advanced | GraphQL, Prisma, TypeScript, WebSockets |