## 12-Week Full-Stack Coding Study Plan (React + Python/Django)

**Time Commitment**: 15–30 min/day on weekdays, 1 hour/day on weekends  
**Goals**: - Learn to build full-stack apps using React and Django. - Be able to spin up simple websites. - Understand basic infrastructure: deployment, databases, backend/frontend architecture.

### Week 1: HTML & CSS + Simple Websites

| Day | Task |
| --- | --- |
| Mon | Learn HTML Basics: [MDN Intro to HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) |
| Tue | Learn CSS Basics: [MDN Intro to CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) |
| Wed | Flexbox and Grid: [Flexbox Froggy](https://flexboxfroggy.com), [CSS Grid Garden](https://cssgridgarden.com) |
| Thu | Build a basic personal profile page |
| Fri | Learn Responsive Design: [FreeCodeCamp Responsive Tutorial](https://www.freecodecamp.org/news/css-media-queries-breakpoints-media-types-standard-resolutions-and-more/) |
| Sat | Project: Create a 1-page website for a fictional local business |
| Sun | Deploy site to [Netlify](https://netlify.com) or [Vercel](https://vercel.com) |

### Week 2: Git + React Basics

| Day | Task |
| --- | --- |
| Mon | Learn Git & GitHub: [Git Handbook](https://guides.github.com/introduction/git-handbook/) |
| Tue | Set up GitHub & push code from VS Code: [GitHub Docs](https://docs.github.com/en/get-started/quickstart) |
| Wed | Intro to React: [React Official Tutorial](https://reactjs.org/tutorial/tutorial.html) |
| Thu | Learn JSX & Components |
| Fri | Learn Props & State |
| Sat | Project: Build a simple React counter or to-do app |
| Sun | Add styling with CSS modules or plain CSS |

### Week 3: React Continued

| Day | Task |
| --- | --- |
| Mon | Learn React Events and Handlers |
| Tue | React Forms: [React Forms Guide](https://reactjs.org/docs/forms.html) |
| Wed | Learn Conditional Rendering |
| Thu | Learn Lists and Keys |
| Fri | Learn useEffect: [useEffect Explained](https://overreacted.io/a-complete-guide-to-useeffect/) |
| Sat | Project: Build a recipe app or book list app |
| Sun | Add responsive design & deploy to Vercel |

### Week 4: Review Week 1

| Day | Task |
| --- | --- |
| Mon | Review HTML/CSS concepts + MDN Docs |
| Tue | Rebuild your profile/landing page from scratch |
| Wed | Review Git & GitHub + Push an old project |
| Thu | Review React basics: components, props, state |
| Fri | Add a feature to one of your React projects |
| Sat | Polish and redeploy a site or app |
| Sun | Rest or explore [Frontend Mentor](https://www.frontendmentor.io/) challenges |

### Week 5: Python Refresher + Django Basics

| Day | Task |
| --- | --- |
| Mon | Review Python Syntax: [W3Schools Python](https://www.w3schools.com/python/) |
| Tue | Learn Virtual Environments: [Python Venv Docs](https://docs.python.org/3/library/venv.html) |
| Wed | Install Django + start project |
| Thu | Understand Django Project vs App Structure |
| Fri | Create Models and Migrations |
| Sat | Create admin panel and test CRUD |
| Sun | Project: Build a blog backend (no frontend yet) |

### Week 6: Django Continued

| Day | Task |
| --- | --- |
| Mon | Views and Templates |
| Tue | URL routing & Namespaces |
| Wed | Static files and template inheritance |
| Thu | Forms and user input |
| Fri | Create authentication system (login/signup) |
| Sat | Project: Add login/auth to your blog |
| Sun | Style your Django templates with CSS |

### Week 7: Django REST Framework (DRF)

| Day | Task |
| --- | --- |
| Mon | Intro to APIs + Install DRF: [DRF Quickstart](https://www.django-rest-framework.org/tutorial/quickstart/) |
| Tue | Build first API View + Serializer |
| Wed | API CRUD for your blog posts |
| Thu | Learn viewsets and routers |
| Fri | Add token authentication |
| Sat | Project: Expose your blog backend as API |
| Sun | Test API with Postman or Insomnia |

### Week 8: Review Week 2 (Django + DRF)

| Day | Task |
| --- | --- |
| Mon | Review Django models and views |
| Tue | Review templates, forms, and auth |
| Wed | Review DRF views, serializers, and routers |
| Thu | Polish and clean up your blog project |
| Fri | Add one feature (e.g. comments or tags) |
| Sat | Deploy your Django app on [Render](https://render.com) or [Railway](https://railway.app) |
| Sun | Rest or write a blog post about what you learned |

### Week 9: Connecting React + Django API

| Day | Task |
| --- | --- |
| Mon | Learn how frontend talks to backend (HTTP requests) |
| Tue | Learn Axios or Fetch: [Axios Docs](https://axios-http.com/) |
| Wed | Set up React app to fetch blog posts from Django API |
| Thu | Create frontend login + token handling |
| Fri | Display fetched data in React (posts, users) |
| Sat | Project: Build blog frontend w/ login, create, update, delete posts |
| Sun | Polish styling and error handling |

### Week 10: Infrastructure, Databases, Deployment

| Day | Task |
| --- | --- |
| Mon | How the Web Works: [MDN Overview](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works) |
| Tue | Learn how databases work (PostgreSQL): [Intro to SQL](https://www.codecademy.com/learn/learn-sql) |
| Wed | How Django connects to PostgreSQL |
| Thu | Learn about env files and config best practices |
| Fri | Learn CI/CD basics: GitHub Actions Overview |
| Sat | Deploy full-stack app (frontend on Vercel, backend on Render/Railway) |
| Sun | Add Google Analytics or a contact form |

### Week 11: Final Full-Stack Project

| Day | Task |
| --- | --- |
| Mon | Plan your app: e.g. Habit Tracker, Notes App, Task Board |
| Tue | Set up backend models and API |
| Wed | Set up frontend project |
| Thu | Build core features end-to-end |
| Fri | Add auth, update, and delete |
| Sat | Polish design + responsiveness |
| Sun | Deploy both ends + write a README |

### Week 12: Review + Showcase

| Day | Task |
| --- | --- |
| Mon | Review full-stack project code |
| Tue | Fix bugs, clean code |
| Wed | Push final version to GitHub |
| Thu | Write a project blog post or case study |
| Fri | Share on LinkedIn, Dev.to, or Twitter |
| Sat | Optional stretch project or hackathon prep |
| Sun | Rest + Reflect on 12 weeks |

### ✅ Recommended Tools

* Code: [VS Code](https://code.visualstudio.com/), [Replit](https://replit.com/)
* Design: [Figma](https://figma.com), [Heroicons](https://heroicons.com)
* Deployment: [Vercel](https://vercel.com), [Render](https://render.com)
* APIs: [Postman](https://postman.com), [Insomnia](https://insomnia.rest)
* Hosting: [GitHub Pages](https://pages.github.com), [Netlify](https://netlify.com)

Let me know if you’d like a printable version (PDF or Google Doc) or if you’d like this plan adapted for Notion or another tool.