PYTHON FULL STACK DEVELOPER

COMPANION RESOURCE KIT

Assess your skills, plan your study time, and get ready for interviews and certification

1. Skill Self-Assessment Matrix

Instead of tracking which topics you've covered, this matrix asks a more useful question: how well do you actually know them? For each skill area, read the three descriptions and be honest about which one fits you today. Revisit this every few weeks, watching a row move from "Beginner" to "Job-Ready" is a far better progress signal than a checklist of topics you've technically "studied."

Skill Area	Beginner	Intermediate	Job-Ready
Python Core	Can follow code but struggles to write a function from scratch	Writes working scripts, but code isn't always clean or reusable	Writes modular, tested code others can read and extend
Front-End (HTML/CSS/JS)	Can build a static page from a tutorial	Can style responsively and handle basic DOM events unaided	Builds interactive UIs and consumes APIs confidently
Databases & SQL	Knows basic SELECT/INSERT syntax	Can design a schema with relationships and write JOINs	Optimizes queries and designs schemas for real applications
Python Framework	Followed a Django/FastAPI tutorial once	Built one working app with models, views, and routes	Can start a new project and structure it without a template
APIs	Understands what an API is conceptually	Has built and tested basic CRUD endpoints	Designs secure, documented APIs consumed by other systems

Git & Collaboration	Uses Git commands when told to	Comfortable with branches and resolving simple conflicts	Manages pull requests and collaborates on shared codebases
Testing & Debugging	Debugs by trial and error	Writes basic unit tests for own code	Has a repeatable process and meaningful test coverage
Deployment & Cloud	Has never deployed an app live	Has deployed at least one project to a live URL	Comfortable with environments, domains, and basic CI/CD

Example: *If your Databases row lands on "Beginner" while everything else is "Intermediate," that's a clear signal to slow down and revisit SQL before starting your next project — rather than pushing forward and hitting the same gap again later.*

2. Curated Free Learning Resources

The roadmap tells you what to learn; this section points you to where you can actually learn it, using resources that are free, current, and widely used by working developers. Bookmark this page rather than searching from scratch every time you start a new stage.

Python Core

- Official Python documentation the most reliable reference for syntax and standard library behavior

- Real Python — practical, example-driven tutorials for intermediate concepts like OOP and decorators
- Python Tutor — visualizes how your code executes line by line, useful when debugging logic errors

Front-End (HTML, CSS, JavaScript)

- MDN Web Docs — the standard reference for HTML, CSS, and JavaScript
- JavaScript.info — a structured, in-depth path through modern JavaScript
- Flexbox Froggy and Grid Garden — short interactive games for practicing CSS layout

Databases & SQL

- PostgreSQL official tutorial — a solid introduction to relational database concepts
- SQLZoo — interactive SQL exercises with instant feedback
- Mode's SQL tutorial — good for practicing JOINs and aggregation on realistic datasets

Django / FastAPI

- Official Django documentation and the "Django Girls" tutorial for a gentler first project
- Official FastAPI documentation — notably clear, with runnable examples for every concept
- Django REST Framework docs — once you're ready to expose your Django app as an API

Git & GitHub

- GitHub Skills — short, hands-on courses that run inside real GitHub repositories
- Official Git documentation — best for understanding what commands actually do under the hood

Testing & Deployment

- pytest official documentation — the standard for Python testing
- Render and Railway free-tier docs — beginner-friendly platforms for deploying your first live project
- Docker's official "Get Started" guide — enough to understand containers without going deep into DevOps

3. Weekly Study Planner Template

Open-ended learning plans tend to quietly stall. A short weekly check-in what you focused on, how much time you actually spent, and what got in the way keeps your progress visible and makes it easier to adjust before you fall too far behind. Copy this table for each week of your journey.

Week	Stage Focus	This Week's Goal	Hours Planned	Hours Actual	Blockers / Notes
1	Python Core	Finish loops & functions; build a calculator	8	6	Struggled with recursion — revisit next week
2					

3					
4					

Example: *Row 1 is filled in as a sample. Notice it records a shortfall (6 of 8 planned hours) and a specific blocker — that's more useful than a vague "studied Python" note, because it tells future-you exactly what to revisit.*

4. Interview Prep: Common Questions by Layer

Once your skills matrix looks solid, shift your practice toward explaining your knowledge out loud interviews test communication as much as correctness. Below are representative questions grouped by layer of the stack; use them for mock interviews or to check whether you can explain a concept without looking it up.

Python Core

- What's the difference between a list and a tuple, and when would you choose one over the other?
- How does Python's garbage collection work, at a high level?
- Explain the difference between *args and **kwargs with a short example.

Front-End

- What's the difference between margin and padding, and how does the box model affect layout?
- How would you fetch data from an API and handle a failed request gracefully?
- What problem do Flexbox and Grid each solve best?

Databases

- What's the difference between an INNER JOIN and a LEFT JOIN?
- When would you choose a NoSQL database over a relational one?
- How would you design a schema for a simple appointment booking system?

APIs & Back-End

- Walk me through what happens between a client request and a server response for a login endpoint.
- How do you handle authentication in a REST API you've built?
- What's the difference between PUT and PATCH?

System Design (entry-level)

- How would you design a basic URL shortener?
- What would you consider if your application's database queries started slowing down under load?

Example: *For each question, practice a two-part answer: a direct explanation in 2–3 sentences, followed by a concrete example from a project you've built. Interviewers consistently rate real examples higher than textbook definitions.*

5. Portfolio & Resume Checklist

A working project and a project that reads well to a recruiter are not the same thing. This checklist covers the presentation layer of your work — the part that decides whether someone clicks into your code at all.

GitHub Profile

- Pin your 3–5 strongest projects — not every repo you've ever created
- Each pinned project has a README with what it does, the tech stack, and a live demo link if available
- Commit history shows regular, incremental commits rather than one large "final upload"
- No exposed API keys, passwords, or .env files in your repositories

Resume Bullet Points

Quantify impact wherever possible, and describe what the project does for a user not just the technologies used.

- Weak: "Built a web app using Django and PostgreSQL."

- Stronger: "Built and deployed a booking platform using Django and PostgreSQL, handling 200+ concurrent appointment records with role-based access for admins and customers."
- Weak: "Created REST API endpoints."
- Stronger: "Designed and documented 12 REST API endpoints consumed by a React front end, reducing average response payload size by restructuring the data model."

Example: *If you don't have real usage metrics yet, describe scope and complexity instead of the number of endpoints, tables in your schema, or user roles supported, rather than leaving the bullet generic.*

6. Certification Readiness Checklist

Certification works best once you have practical experience to back it up it validates skills you already have rather than replacing the process of building them. Use this short self-check to gauge your readiness.

- You can build a CRUD application end-to-end without following a tutorial step by step
- You're comfortable explaining your own project's architecture out loud, not just running it
- You've deployed at least one project so it's reachable outside your own machine
- You understand basic security practices well enough to avoid the most common mistakes (plaintext passwords, exposed credentials)
- You want a credential that signals your skills clearly to recruiters who don't have time to review your full GitHub history

If most of these are true, a structured certification can help package your self-taught skills into something recruiters can verify quickly rather than asking them to take your word for it.

Ready to Make It Official?

This kit is designed to sharpen the skills and presentation you've built through the roadmap. When you're ready to back that up with a recognized, verifiable credential, GSDC's certification is built specifically for developers at this stage.

Explore the **GSDC Certified Python Full Stack Developer Certification** and take the next step in your full-stack development career.

GSDC
Global Skill Development Council

CERTIFIED PYTHON FULL STACK DEVELOPER (CFSD)

BUILD PYTHON FULL STACK EXPERTISE ACROSS FRONT-END, BACK-END, DATABASES, APIS, AND MODERN FRAMEWORKS TO CREATE SCALABLE WEB APPLICATIONS.

ABOUT GSDC CERTIFICATION

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.

LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Demonstrate front-end as well as back-end technical skills.
- Analysis of Integrating database with other resources such as APIs and Cloud integration technology.
- Verify understanding of frameworks in Java and Python.

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now

www.gsdcouncil.org