Software Development

Program Brochure · 2026

SYSTEM DESIGN & ARCHITECTURE

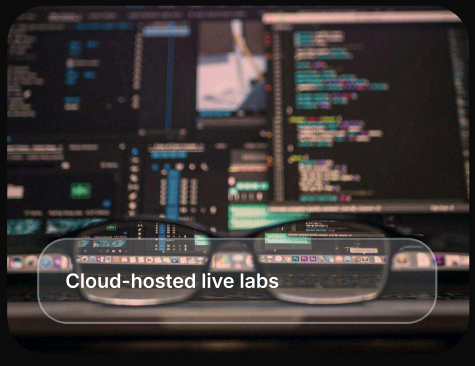
Design systems that scale.

Advanced ↗ | 64 Hours ↗ | SAM-SYSDESIGN ↗

Software Development

Code that ships.

18 programs in this track



7

modules · hands-on labs · capstone project

Inside this brochure

Contents

Program overview 03

Who it's for 04

Learning outcomes 05

Program runway 06

Detailed curriculum 07

Tools & frameworks 10

Projects & capstone 11

Certification & method 12

Learning path 13

FAQs 14



AT A GLANCE

Level	**Advanced**
Duration	**64 Hours**
Modules	**7**
Certificate	**SAM-SYSDESIGN**

Program overview

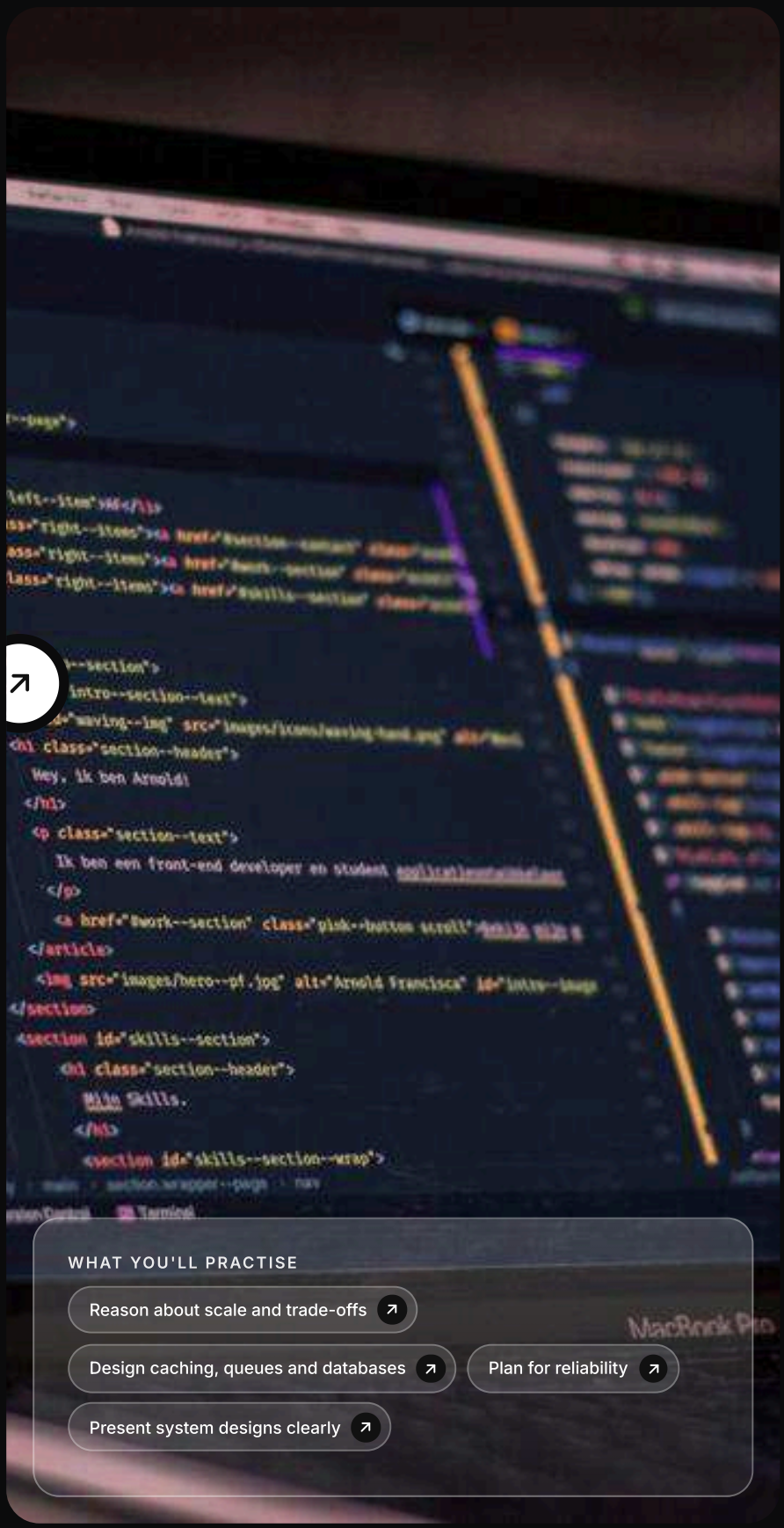
Design systems that scale.

Design scalable, reliable systems — trade-offs, components and real-world case studies.

Designing systems that stay fast and reliable as usage grows is a core skill for senior engineers and a key part of technical interviews. This advanced program builds structured design thinking: requirements and estimation, scaling, databases, caching, queues, consistency, reliability and observability, applied through real-world case studies and regular design reviews.

DELIVERY

ILT · VILT · Blended

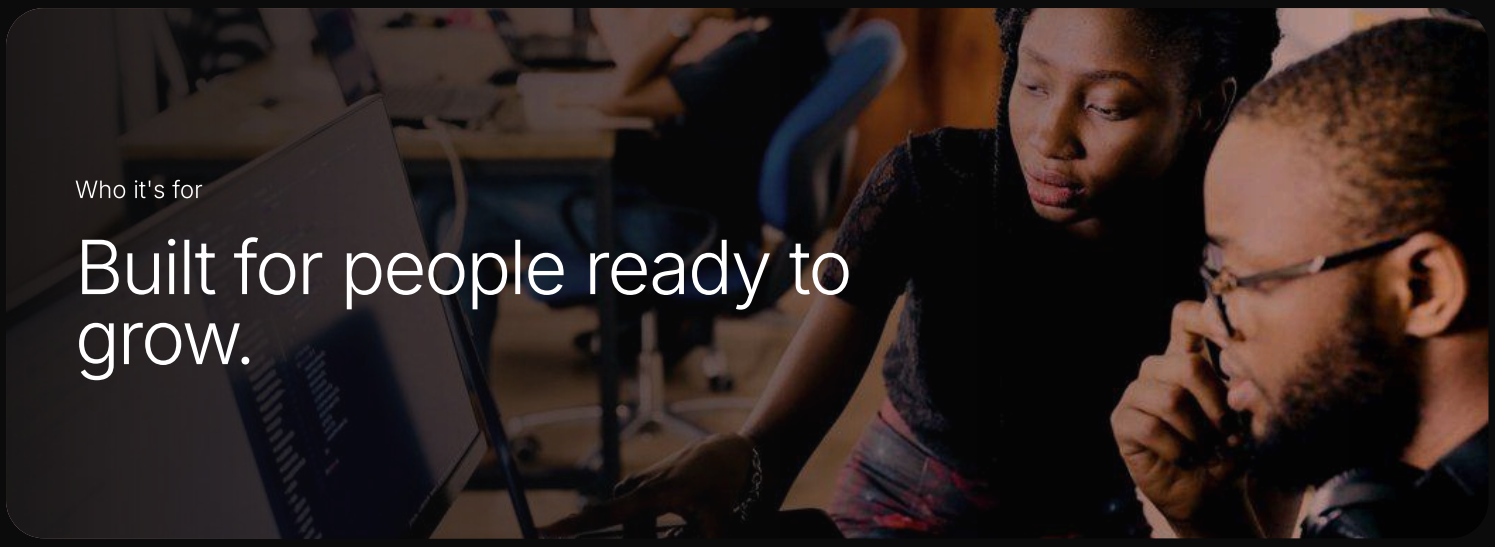


LEVEL
Advanced

DURATION
64 Hours

CERTIFICATION
SAM-SYSDESIGN

TRACK
Software Development



Who it's for

Built for people ready to grow.

01

Students & freshers

Python, Java, C/C++ and JavaScript foundations

02

Frontend developers

React, Angular and TypeScript

03

Backend developers

Node, Django, FastAPI and Spring Boot

04

Senior engineers

Microservices, system design, leadership

PREREQUISITES

- Experience building and deploying web applications 01
- Working knowledge of databases and APIs 02
- Familiarity with basic networking concepts 03

ROLES THIS PREPARES YOU FOR

- Senior Software Engineer ↗
- Backend Engineer ↗
- Software Architect (aspiring) ↗
- Tech Lead ↗
- Engineers preparing for system design interviews ↗

Learning outcomes

What you'll be able to do.



01	Clarify requirements and estimate capacity for system designs
02	Design scalable architectures using load balancing and partitioning
03	Choose appropriate SQL, NoSQL and storage technologies
04	Apply caching, queues and streaming to improve performance
05	Plan for reliability, consistency and graceful failure handling
06	Communicate designs clearly with diagrams and documented trade-offs

Program runway

Module by module.

7
modules · 64 hours

01	**Design Thinking and Fundamentals** Functional and non-functional requirements, Back-of-envelope estimation, Latency, throughput and availability	8 hrs
02	**Scalability and Load Balancing** Vertical versus horizontal scaling, Load balancers and algorithms, Stateless services and sessions	8 hrs
03	**Databases and Storage** SQL versus NoSQL selection, Indexing and query performance, Replication and read replicas	10 hrs
04	**Caching and Asynchronous Processing** Cache-aside, write-through and write-back, Redis and cache invalidation, Message queues and pub/sub	10 hrs
05	**Consistency and Distributed Systems** CAP and PACELC theorems, Consistency models, Leader election and consensus overview	8 hrs
06	**Reliability and Observability** Redundancy and failover, Disaster recovery, RPO and RTO, SLIs, SLOs and error budgets	8 hrs
07	**Case Studies and Design Reviews** Design a URL shortener, Design a news feed, Design a chat system	12 hrs
★	**Capstone · End-to-End Platform Design Review** Assessment, capstone review and certification	

The syllabus.

Modules 01–03

Module 01

Design Thinking and Fundamentals

8 hours

Hands-on lab
Estimate capacity and draft requirements for a URL shortener.

TOPICS COVERED

- Functional and non-functional requirements
- Back-of-envelope estimation
- Latency, throughput and availability
- Client-server, DNS and CDNs
- A structured design framework

Module 02

Scalability and Load Balancing

8 hours

Hands-on lab
Design and diagram a horizontally scalable API tier.

TOPICS COVERED

- Vertical versus horizontal scaling
- Load balancers and algorithms
- Stateless services and sessions
- API gateways and rate limiting
- Consistent hashing

Module 03

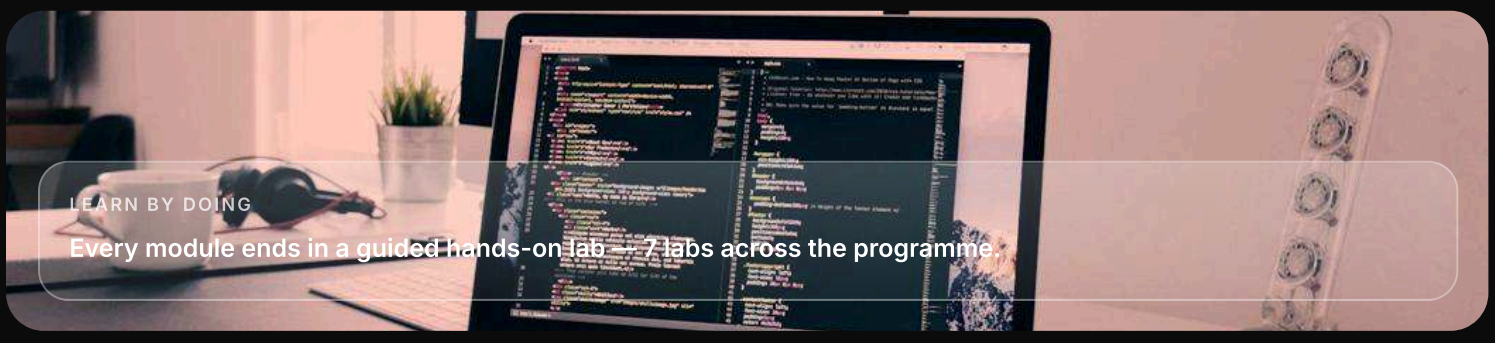
Databases and Storage

10 hours

Hands-on lab
Choose and justify a data model and sharding scheme.

TOPICS COVERED

- SQL versus NoSQL selection
- Indexing and query performance
- Replication and read replicas
- Sharding and partitioning strategies
- Object and blob storage



LEARN BY DOING

Every module ends in a guided hands-on lab — 7 labs across the programme.

Syllabus, continued.

Modules 04–05

Module 04

Caching and Asynchronous Processing

10 hours

Hands-on lab
Add caching and a queue to a slow order-processing design.

TOPICS COVERED

- Cache-aside, write-through and write-back
- Redis and cache invalidation
- Message queues and pub/sub
- Event streaming with Apache Kafka
- Backpressure and idempotency

Redis · Apache Kafka

Module 05

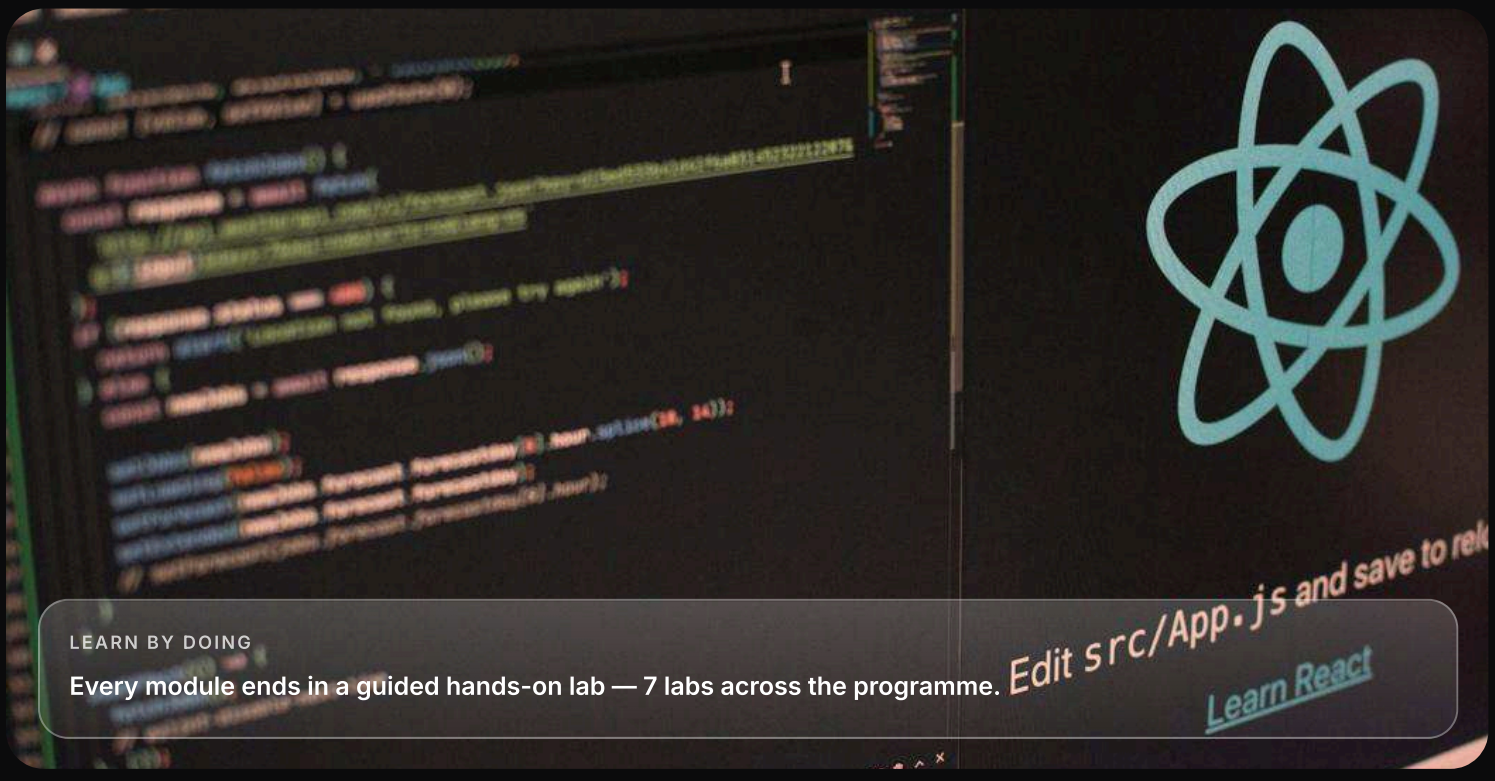
Consistency and Distributed Systems

8 hours

Hands-on lab
Analyse consistency trade-offs for a payments ledger design.

TOPICS COVERED

- CAP and PACELC theorems
- Consistency models
- Leader election and consensus overview
- Distributed transactions and sagas
- Clocks, ordering and idempotency keys



LEARN BY DOING

Every module ends in a guided hands-on lab — 7 labs across the programme.

Syllabus, continued.

Modules 06–07

Module 06

Reliability and Observability

8 hours

Hands-on lab
Define SLOs and a failure-recovery plan for a chosen system.

TOPICS COVERED

- Redundancy and failover
- Disaster recovery, RPO and RTO
- SLIs, SLOs and error budgets
- Logging, metrics and tracing
- Security by design basics

Module 07

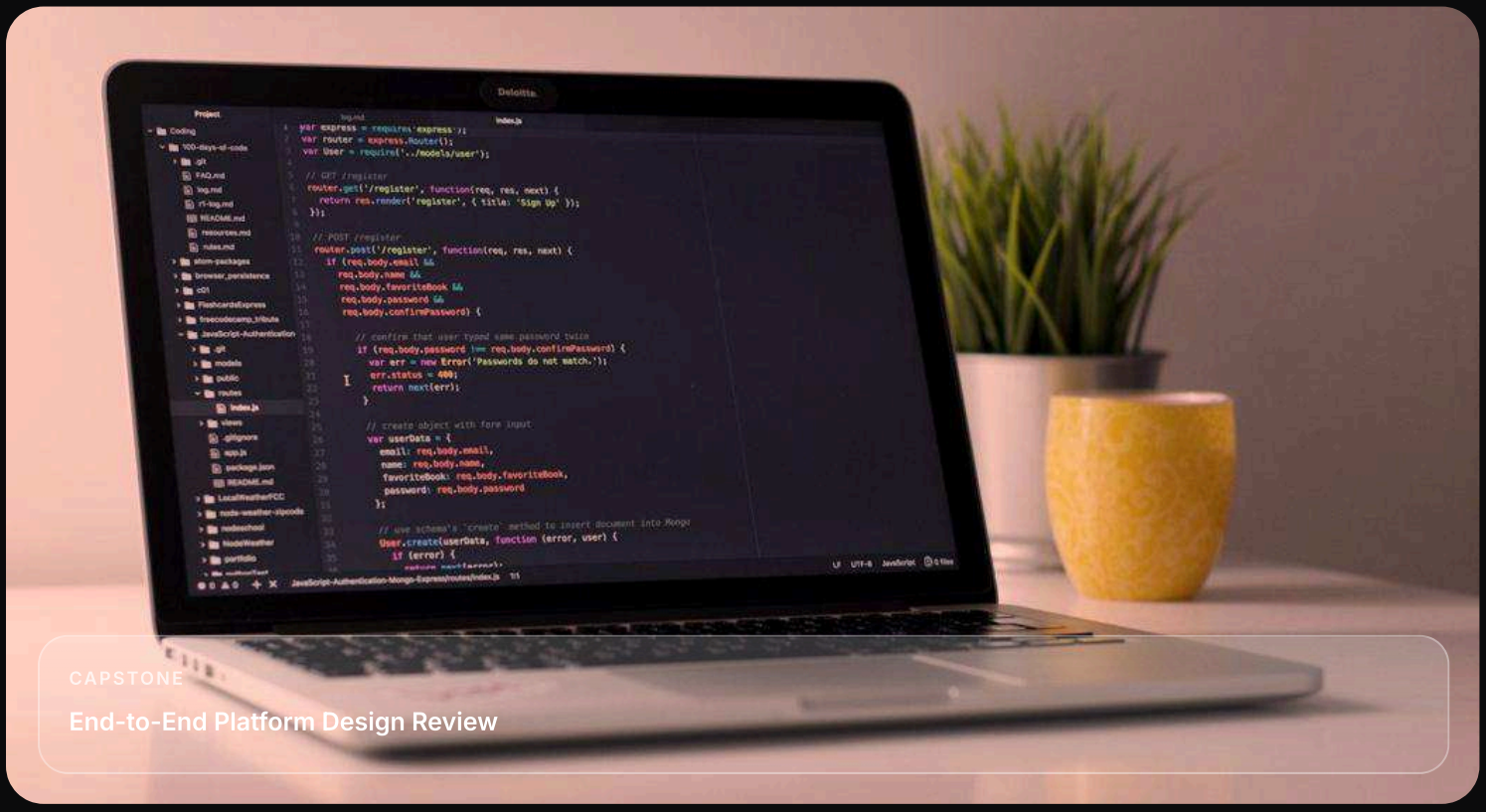
Case Studies and Design Reviews

12 hours

Hands-on lab
Present a full design and defend trade-offs in peer review.

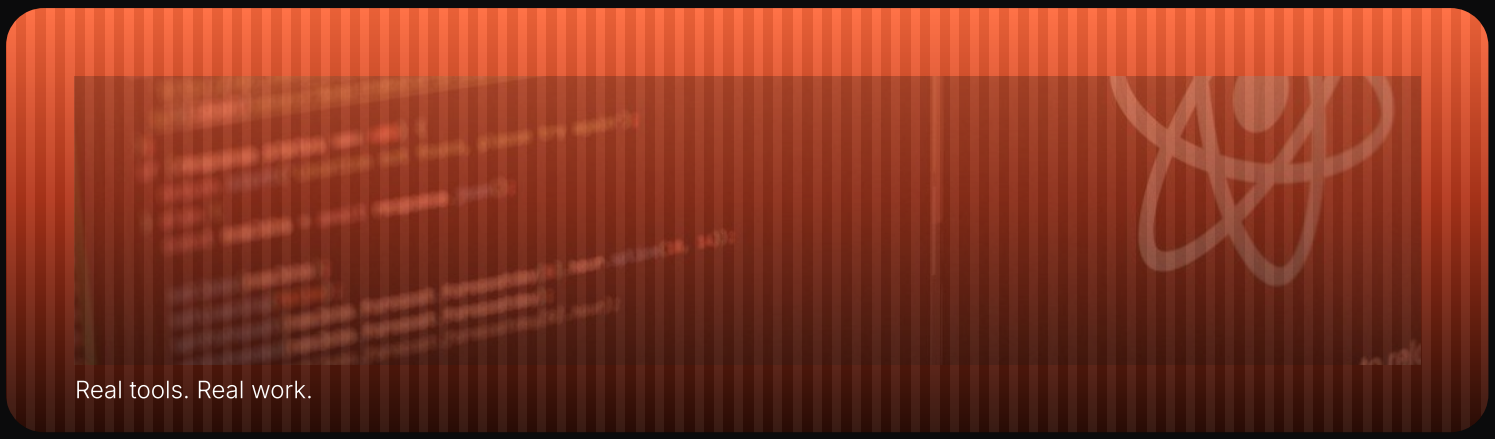
TOPICS COVERED

- Design a URL shortener
- Design a news feed
- Design a chat system
- Design a video streaming service
- Design a ride-hailing service
- Design reviews and interview practice



CAPSTONE

End-to-End Platform Design Review



Redis
In-memory data store

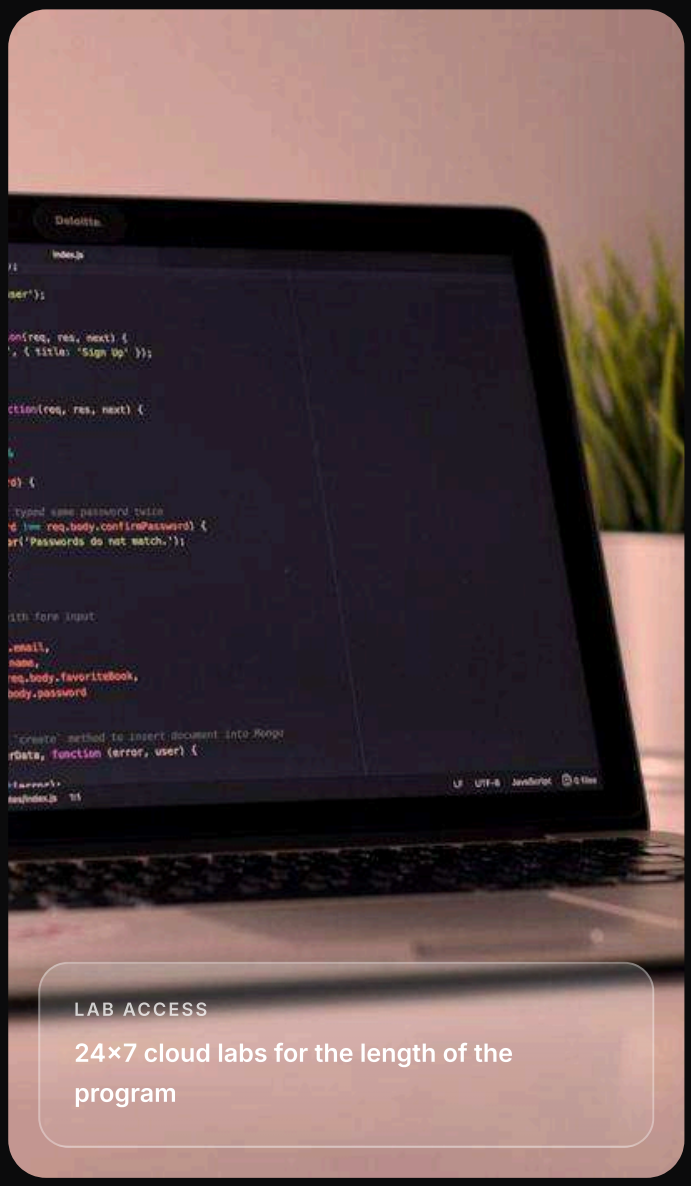
PostgreSQL
Relational database

Apache Kafka
Event streaming

Docker
Containers & images

What you will build

Reason about scale and trade-offs	01
Design caching, queues and databases	02
Plan for reliability	03
Present system designs clearly	04



LAB ACCESS

24×7 cloud labs for the length of the program

01

Scalable URL Shortener

Design and prototype a URL shortener with capacity estimates, key generation, caching with Redis and analytics pipeline.

02

Real-Time Chat Design

Architect a chat system covering WebSocket connections, message storage, delivery guarantees, presence and horizontal scaling.

03

Feed System Prototype

Design a social news feed comparing fan-out on write and read, with caching and queue-based processing.



Capstone project

End-to-End Platform Design Review

Produce a complete design for a large-scale system such as food delivery or video streaming, including requirements, estimates, architecture diagrams, data models,

Certification

SAM-SYSDESIGN

Awarded on successful completion of the program assessments.

- MCQ assessments
- Practical lab evaluation
- Project & capstone review
- Final presentation



The SAM Learning Framework™

Scenario · Apply · Master · Mentor

Phase 01

Scenario

Real-world attack and use-case scenarios open every module.

Phase 02

Apply

Hands-on lab work, CTF challenges and live projects.

Phase 03

Master

Assessment, peer review and certification prep.

Phase 04

Mentor

Post-program career guidance and support.

DELIVERY MODELS

- Instructor-Led (ILT)
- Virtual Instructor-Led
- Self-Paced Online
- E-Learning Licensing
- Blended Learning
- Corporate Onsite
- Bootcamp Intensive

Learning path · Software Development

Where this program fits.

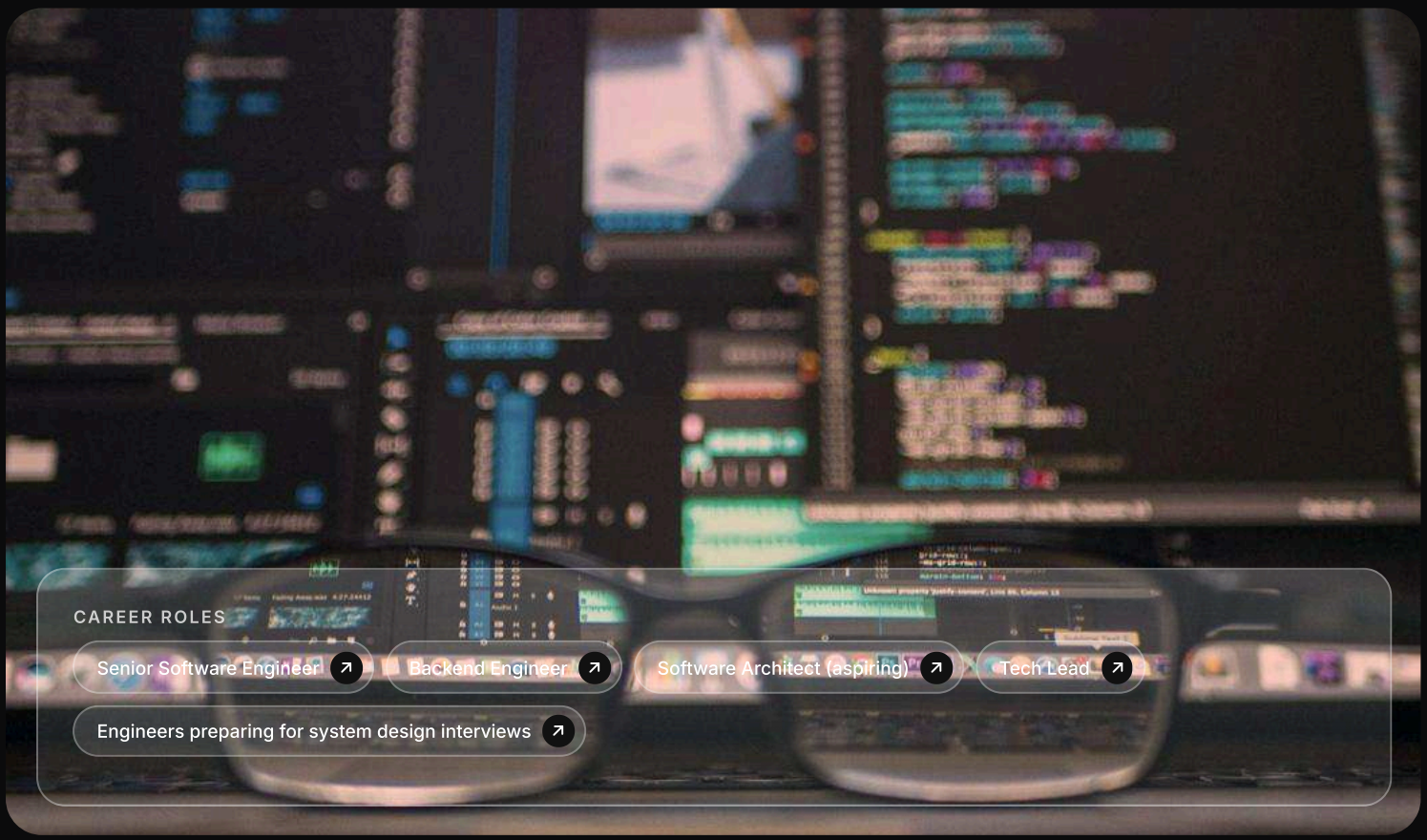
Beginner	Intermediate	Advanced	Expert
4 programs	10 programs	You are here	1 programs

Before this

Python Advanced (OOP, Design Patterns)	56 Hours
TypeScript Mastery	40 Hours
Django Web Framework	56 Hours
FastAPI — Modern Python APIs	48 Hours

Up next

Software Engineering Leadership	48 Hours



Questions, answered.

Q01

Do I need prior experience to join?

Yes. This advanced program assumes you have built and deployed applications and understand databases and APIs. It suits working engineers and those preparing for senior roles or system design interviews.

Q02

How are classes scheduled and delivered?

Sessions are instructor-led and run as live virtual or blended batches, with weekday and weekend options where available. Each session combines concept walkthroughs, live coding and guided lab time, with recordings or notes shared for revision.

Q03

Will I receive a certificate?

Yes. Learners who complete the required labs, projects and final assessments receive the Samcommunity Academy SAM-SYSDESIGN certificate. It is awarded on the basis of assessed work, not attendance alone.

Q04

What tools and lab access will I need?

You mainly need a laptop with a diagramming tool such as Excalidraw or draw.io, plus Docker for optional prototypes using Redis, PostgreSQL and Kafka. All tools are free.

Q05

Is career support included?

You get guidance on building a portfolio from your projects, resume and GitHub profile reviews, and mock technical interviews. We help you prepare and present your skills, but we do not guarantee jobs, placements or salary outcomes.



Let's talk

START SYSTEM DESIGN & ARCHITECTURE.

academy@samcommunity.in

corporate@samcommunity.in

samcommunity.in

@samcommunity_

Learn practical skills. Grow with mentors.

Xylyn Samcommunity Private Limited · DPIIT-recognized startup · Surat, Gujarat, India