

Kabir Mehta

Backend Developer · FICTIONAL SAMPLE
Ahmedabad, India

Skills

Node.js

Applied to settlement reconciliation api through documented preparation, execution and review.

NestJS

Applied to multi-tenant access service through documented preparation, execution and review.

TypeScript

Applied to background job recovery through documented preparation, execution and review.

PostgreSQL

Applied to settlement reconciliation api through documented preparation, execution and review.

Redis

Applied to multi-tenant access service through documented preparation, execution and review.

message queues

Applied to background job recovery through documented preparation, execution and review.

API contracts

Applied to settlement reconciliation api through documented preparation, execution and review.

observability

Applied to multi-tenant access service through documented preparation, execution and review.

Certifications

Distributed systems design studio

Meridian Professional Learning Studio (fictional) 2023 continuing learning - not a licence or certification Node.js and postgresql in the context of backend developer work. Used reflective exercises and a bounded practice example related to settlement reconciliation api.

Summary

Fictional demonstration candidate. All employers, education, projects, achievements and outcomes are illustrative and unverified. Not a real application or licence record. Backend Developer with an illustrative 11-year career in product engineering and enterprise applications. Combines node.js, nestjs, typescript with disciplined documentation, practical coordination and clear communication. The sample career progresses from focused execution to independent workstream ownership, with responsibilities and boundaries described for each appointment. Selected work includes settlement reconciliation api, multi-tenant access service and background job recovery. These examples explain the original problem, individual contribution, deliverables, review approach and remaining limitations rather than relying on unsupported headline claims. Prepared for experienced backend developer opportunities requiring dependable delivery, thoughtful professional judgement and collaboration. The qualification narrative includes M.Tech in Computer Science. Every named organisation and outcome in this record is fictional; professional eligibility is not independently established. Turn a clear brief into dependable product engineering and enterprise applications work: understand the context, apply node.js and nestjs, record the evidence and explain the limitations before handover.

Experience

Backend Developer - Meridian Digital Systems (fictional)

2024-07 - Present

Independent ownership of scoped backend developer work, coordinating contributors and making review requirements explicit. Includes the first two selected work examples. Introduced idempotency keys and an auditable reconciliation state machine. Applied scoped repositories and isolation test fixtures. Led brief clarification and prioritised work using node.js, nestjs and typescript. Raised unresolved constraints before committing to the next stage. Coordinated peer reviews and handover preparation; used a architecture decision record to distinguish completed work, assumptions and follow-up needs. Supported colleagues with practical examples of postgresql and maintained a concise learning record after important assignments. Illustrative career responsibilities. Employer confirmation and underlying work records are not supplied.

Senior Backend Developer - Northline Digital Systems (fictional)

2020-07 - 2024-06

Owned defined assignments and supported cross-functional coordination. Developed deeper practice in typescript and postgresql. Implemented retry queues and an operator replay console. Translated incoming requirements into a sequenced plan and aligned responsibilities with the project or service owner. Applied redis and message queues to resolve delivery questions while maintaining source and decision notes. Introduced reusable working documents and reviewed exceptions with the responsible specialist rather than silently changing scope. Prepared a test and release checklist so the next team could understand the work and remaining questions. Illustrative career responsibilities. Employer confirmation and underlying work records are not supplied.

Backend Developer - Cedarbridge Digital Systems (fictional)

2017-07 - 2020-06

Progressed from supported tasks to independently managed assignments, with review available for unfamiliar or higher-risk decisions. Handled recurring work involving node.js and nestjs using a documented preparation and review process. Supported multi-tenant access service by organising inputs, maintaining issue notes and incorporating reviewer feedback. Coordinated colleagues and internal stakeholders using concise status updates, clear questions and agreed next steps. Improved record consistency through api contracts and documented handover expectations. Illustrative career responsibilities. Employer confirmation and underlying work records are not supplied.



Kabir Mehta

Backend Developer · FICTIONAL SAMPLE
Ahmedabad, India

Secure application delivery workshop

Meridian Professional Learning Studio (fictional) 2024 continuing learning - not a licence or certification Nestjs and redis in the context of backend developer work. Used reflective exercises and a bounded practice example related to multi-tenant access service.

Accessible interface review

Meridian Professional Learning Studio (fictional) 2025 continuing learning - not a licence or certification Typescript and message queues in the context of backend developer work. Used reflective exercises and a bounded practice example related to background job recovery.



Languages

English

professional working - fictional sample

Hindi

professional working - fictional sample

Associate Backend Developer - Cedarbridge Digital Systems (fictional)

2015-07 - 2017-06

Built practical foundations through supervised assignments, routine documentation and feedback from experienced colleagues. Assisted with node.js and typescript within an agreed scope and escalated unfamiliar work. Prepared inputs and checked completeness before passing work to the responsible reviewer. Maintained task records and learned to communicate assumptions, constraints and observed problems clearly. Applied review feedback to subsequent assignments and developed a dependable working routine. Illustrative career responsibilities. Employer confirmation and underlying work records are not supplied.



Education

M.Tech in Computer Science - Asterbridge Institute of Professional Studies (fictional institution)

2013-07 - 2015-05

completed - fictional record Node.js NestJS PostgreSQL Specialist study on settlement reconciliation api; an illustrative learning project, not a published result.

B.Tech in Computer Science - Cedarhaven College of Applied Studies (fictional institution)

2009-07 - 2013-05

completed - fictional record TypeScript Redis message queues Applied coursework in node.js, documentation and reviewed practical assignments.



Projects

Settlement reconciliation API

Fictional internal work programme in product engineering and enterprise applications; not a real client case study. Problem: Retries could create ambiguous payment records. Objective: Create a workable response to this issue through node.js, explicit review criteria and practical documentation. Agree the boundaries before execution and retain unresolved points for follow-up. Backend Developer; owned the stated workstream, not the full organisation or every collaborator contribution. Introduced idempotency keys and an auditable reconciliation state machine. Prepared the scope with the commissioning team, identified unresolved inputs and used node.js to turn the brief into a sequenced work package. Applied nestjs and typescript while coordinating reviews with the designated owner. Kept decision notes so collaborators could separate facts, assumptions and changes. Assembled the handover material, explained open limitations and agreed which items needed further review rather than presenting them as completed. Methods: Node.js; NestJS; TypeScript; PostgreSQL Deliverables: Settlement reconciliation API - scoped brief; Settlement reconciliation API - reviewed working package; Settlement reconciliation API - handover and learning summary Review: Reviewed the scoped output against the agreed brief, recorded exceptions and checked that key conclusions could be traced to observations. The review package calls for a architecture decision record and a named human reviewer. Illustrative outcome: the team adopted a repeatable approach for settlement reconciliation api, with clearer ownership and reviewable records. This is a fictional qualitative result; no real performance measurement or external acceptance evidence is supplied. Limitations: Synthetic demonstration only. API boundaries, recovery behaviour, access control and release ownership are documented with each work example. No underlying client documents, independently verified measurements or signed approval records are attached.

Multi-tenant access service

Fictional internal work programme in product engineering and enterprise applications; not a real client case study. Problem: Tenant boundaries were inconsistent in queries. Objective: Create a workable response to this issue through nestjs, explicit review criteria

Kabir Mehta

Backend Developer · FICTIONAL SAMPLE

Ahmedabad, India

and practical documentation. Agree the boundaries before execution and retain unresolved points for follow-up. Backend Developer; owned the stated workstream, not the full organisation or every collaborator contribution. Applied scoped repositories and isolation test fixtures. Prepared the scope with the commissioning team, identified unresolved inputs and used typescript to turn the brief into a sequenced work package. Applied postgresql and redis while coordinating reviews with the designated owner. Kept decision notes so collaborators could separate facts, assumptions and changes. Assembled the handover material, explained open limitations and agreed which items needed further review rather than presenting them as completed. Methods: NestJS; TypeScript; PostgreSQL; Redis Deliverables: Multi-tenant access service - scoped brief; Multi-tenant access service - reviewed working package; Multi-tenant access service - handover and learning summary Review: Reviewed the scoped output against the agreed brief, recorded exceptions and checked that key conclusions could be traced to observations. The review package calls for a reviewed pull-request summary and a named human reviewer. Illustrative outcome: the team adopted a repeatable approach for multi-tenant access service, with clearer ownership and reviewable records. This is a fictional qualitative result; no real performance measurement or external acceptance evidence is supplied. Limitations: Synthetic demonstration only. API boundaries, recovery behaviour, access control and release ownership are documented with each work example. No underlying client documents, independently verified measurements or signed approval records are attached.

Background job recovery

Fictional internal work programme in product engineering and enterprise applications; not a real client case study. Problem: Failed notifications disappeared from operations view. Objective: Create a workable response to this issue through typescript, explicit review criteria and practical documentation. Agree the boundaries before execution and retain unresolved points for follow-up. Backend Developer; owned the stated workstream, not the full organisation or every collaborator contribution. Implemented retry queues and an operator replay console. Prepared the scope with the commissioning team, identified unresolved inputs and used redis to turn the brief into a sequenced work package. Applied message queues and api contracts while coordinating reviews with the designated owner. Kept decision notes so collaborators could separate facts, assumptions and changes. Assembled the handover material, explained open limitations and agreed which items needed further review rather than presenting them as completed. Methods: TypeScript; PostgreSQL; Redis; message queues Deliverables: Background job recovery - scoped brief; Background job recovery - reviewed working package; Background job recovery - handover and learning summary Review: Reviewed the scoped output against the agreed brief, recorded exceptions and checked that key conclusions could be traced to observations. The review package calls for a test and release checklist and a named human reviewer. Illustrative outcome: the team adopted a repeatable approach for background job recovery, with clearer ownership and reviewable records. This is a fictional qualitative result; no real performance measurement or external acceptance evidence is supplied. Limitations: Synthetic demonstration only. API boundaries, recovery behaviour, access control and release ownership are documented with each work example. No underlying client documents, independently verified measurements or signed approval records are attached.

Achievements

Settlement reconciliation API

Introduced idempotency keys and an auditable reconciliation state machine. The achievement is the described workstream contribution; independent outcome evidence is not supplied.

Multi-tenant access service

Applied scoped repositories and isolation test fixtures. The achievement is the described workstream contribution; independent outcome evidence is not supplied.



Kabir Mehta

Backend Developer · FICTIONAL SAMPLE
Ahmedabad, India

Background job recovery

Implemented retry queues and an operator replay console. The achievement is the described workstream contribution; independent outcome evidence is not supplied.

