

## Experience

---

### Backend Developer

#### CloudTechTiq

Nov 2024 – Present

- Architected a multi-tenant hosting and billing/ERP SaaS on **Django REST Framework** spanning 25+ apps and 230+ ViewSets under a unified `/api/` namespace.
- Designed a dual-gateway payment system (**Razorpay + Stripe**) with HMAC-SHA256 webhook idempotency to prevent duplicate charges, coordinating payment, invoicing, provisioning, and double-entry ledger reconciliation, plus Indian GST e-invoicing (EINV-2.1) through the NIC gateway with IRN generation and signed QR codes.
- Secured the platform with RS512 **JWT** auth (kid-keyed JWKS, refresh-token rotation over HTTPOnly cookies), a 2FA flow over email OTP and RFC-6238 TOTP, and a 43-permission department-scoped RBAC.
- Shipped real-time notifications over a **Django Channels 4 + Redis WebSocket** server with JWT-authenticated async consumers, backed by a 24-task **Celery** system (late-ack, worker-lost requeue).
- Eliminated N+1 queries via Prefetch/OuterRef/Exists across a dual async/sync database layer (asynccpg + psycopg) on SQLAlchemy 2.0 with connection pooling and keyset cursor pagination.
- Built a service-provisioning engine that tracks each order through its lifecycle across seven service types, automating bare-metal setup over an HPE iLO Redfish wrapper and Celery-driven IP allocation.
- Set up end-to-end observability with **OpenTelemetry** tracing and a Prometheus/Grafana/Loki/Tempo stack, so failures across services could be traced to a single request.
- Built and open-sourced a type-safe Namecheap SSL/domain SDK (**httpx, Pydantic v2, mypy-strict**), published to PyPI — [github.com/alzywelzy/namecheap-wrapper](https://github.com/alzywelzy/namecheap-wrapper).
- Containerized and shipped the platform through multi-stage Docker images and a GitHub Actions pipeline publishing to GHCR.

### Backend Developer

#### Radixlink

Jun 2023 – Sep 2024

- Built Rosterly, a multi-tenant workforce/payroll SaaS (**Django 4.2 + FastAPI**, 16 apps, 237 REST endpoints) with schema-per-tenant isolation via **django-tenants**.
- Developed the timesheet → payroll → invoicing lifecycle with atomic multi-assignment timesheet processing and automated payroll reconciliation on approval, eliminating N+1s on large payroll runs.
- Integrated **Plaid** bank-account aggregation with transaction sync and duplicate filtering across all tenant schemas, scheduled via **Celery Beat**.
- Modeled polymorphic assignments (C2C/FTE/T&M) with date-range conflict detection and a temporal pay-rate system with effective-date ranges and overlap validation.
- Generated multi-format financial reporting (PDF/Excel/CSV) with WeasyPrint and Pandas, plus transaction-protected sequential PDF invoice numbering.

## Education

---

### Bachelor of Computer Applications (BCA)

#### Bundelkhand University

Jul 2021 – Jun 2024

- Built an AI-powered news summarizer in Python as a final-year academic project, using the OpenAI API to condense full-length articles into short, readable digests.

## Technical Skills

---

- **Languages:** Python, SQL, TypeScript, JavaScript, Bash
- **Backend:** Django, Django REST Framework, FastAPI, Celery, SQLAlchemy, Pydantic, asyncio, WebSockets, SSE, REST, GraphQL
- **Data & Infra:** PostgreSQL, Redis, MongoDB, MySQL, Alembic, RabbitMQ; AWS (EC2, S3, RDS, Lambda), Azure, Apache CloudStack, Docker, Kubernetes, GitHub Actions, GHCR, Gunicorn, NGINX, Linux
- **Security & Payments:** OAuth2/OpenID, JWT, RBAC, MFA, 2FA/TOTP, HMAC, Stripe, Razorpay, Plaid
- **AI, Observability & Testing:** OpenAI API, LangChain; OpenTelemetry, Prometheus, Grafana, Sentry, pytest, TDD, Unit/Integration/E2E, CI/CD, Agile/Scrum