Artem S.

Full Stack Developer

Summary of Qualifications

I build and evolve production software where correctness, performance, and reliability matter, with 3.5 years of experience working on complex backend systems, real-time processing, and SaaS platforms that operate under constant load and changing conditions.

I have worked on automated betting systems, AI-driven recruitment platforms, and long-running ticketing products, taking responsibility for core services, system architecture, and production stability. I enjoy untangling complex business logic, designing clear system boundaries, and turning early-stage solutions into maintainable, scalable products.

I work comfortably across backend and frontend, contribute to architectural decisions, and focus on system stability, performance, and maintainability.

Skills

Programming Languages/Technologies

- Python
- JavaScript/TypeScript
- HTML/CSS/LESS/Sass
- SQL
- WebSockets
- Microservice architecture
- Asynchronous programming
- Technical Documentation/Architecture Reports

AI / Search / Data Processing

- Vector embeddings
- Vector databases (pgvector)
- RAG pipelines
- AI-powered document analysis
- Prompt engineering

RDBMS

- MySQL
- PostgreSQL
- SQLite
- Alembic

NoSQL

- MongoDB
- Firestore

Observability & Reliability

- Prometheus
- Grafana
- Sentry
- Elasticsearch

Virtualization Tools

- Docker
- Kubernetes

CI/CD

- GitHub Actions (CI/CD)
- Jenkins

Frameworks/Libraries

- FastAPI
- Django/DRF
- Flask
- React/Next.js
- SQLAlchemy
- Numpy/Pandas
- Tailwind
- Socket.IO
- OpenAI

Messaging, Streaming & Orchestration

- Kafka
- Redis (queues, pub/sub)
- Temporal Workflows
- Celery

Networking

- Proxy / SOCKS5 integration

Application/Web Servers

- Nginx

Cloud Providers

- AWS
- GCP

Testing Tools

- Postman
- Pytest
- Unitest, Mock

Version Control

- Git
- GitHub
- GitLab

Experience

Real-Time Automated Betting Platform

Project Description: Internal high-performance betting platform designed to automate real-time betting operations within the betting domain. The system was built using a microservices architecture and focused on low-latency, fault-tolerant decision-making, as betting odds and conditions changed dynamically in real time.
The platform consisted of multiple interconnected services, including data ingestion and parsing services, automated bot services for placing bets on external platforms, and strategy-driven decision-making services responsible for selecting optimal betting actions.

Domain: High-Frequency, Real-Time Betting Systems

Involvement Duration: 1 year

Project Role: Full-Stack Developer

Responsibilities:

- Designed, implemented, and maintained bet decision-making services responsible for selecting betting strategies and execution accounts in real time.
- Owned the architecture and performance optimization of betting workflows with a strong focus on low latency and high throughput.
- Developed and supported multiple betting-related microservices involved in odds processing and bet execution.
- Maintained and extended the core matching service, responsible for processing betting events and distributing them via WebSocket in real time.
- Implemented WebSocket-based event delivery between services for real-time betting updates.
- Designed and maintained a proxy management service, including SOCKS5 proxy handling and lifecycle management.
- Ensured fault tolerance and production stability, handled incidents, and monitored services using Sentry, Prometheus, and Grafana.
- Added unit and integration tests and integrated them into the CI/CD deployment pipeline.
- Participated in production support and continuous delivery of new betting features and strategy updates.

Project Team Size: 30-50 team members

Tools & Technologies: FastAPI, SQLAlchemy, Alembic, Docker, Docker Compose, Redis, Celery, MSSQL, socks5, WebSocket, Sentry, Prometheus, Grafana, Elasticsearch

AI-Powered Recruitment Lead Generation Platform

Project Description: This project is a lead-generation SaaS recruitment platform with AI-powered data extraction and analysis capabilities. The system was designed using a microservices architecture and focused on automated processing of customer-provided data.
The platform enabled ingestion, processing, and analysis of structured and unstructured data, supporting scalable and automated recruitment workflows.

Domain: Lead Generation, SaaS, AI

Involvement Duration: 1.5 years

Project Role: Full-Stack Developer

Responsibilities:

- Designed and developed the initial MVP as a monolithic system, covering backend, asynchronous processing, and frontend from scratch.
- Implemented asynchronous data processing pipelines using Celery for parsing and analyzing Excel, PDF, and Word documents with Pandas and AI-based extraction.
- Owned frontend development for the MVP and production system, delivering a functional UI and iterating based on early customer feedback.
- Led the system evolution from a monolithic architecture to a microservices-based platform after successful MVP validation.

- Designed and implemented the core orchestration service responsible for coordinating data ingestion, processing, and AI-driven workflows across multiple services.
- Migrated critical business workflows from Celery to Temporal to improve fault tolerance, retry semantics, and long-running workflow reliability.
- Solved complex architectural challenges related to replacing task-based execution with workflow-based orchestration while preserving existing business logic.
- Integrated Kafka as an event backbone and implemented pub/sub communication between services.
- Designed and implemented interoperability between Kafka and Temporal to enable event-driven workflow execution and cross-service coordination.
- Ensured system scalability, reliability, and observability during the transition to a distributed architecture.
- Participated in production support, debugging distributed workflows, and continuous delivery of new features.

Project Team Size: 5 - 10 team members

Tools & Technologies: FastAPI, SQLAlchemyl, Alembic, Docker, Docker Compose, Redis, Celery, Temporal, Kafka, PostgreSQL, WebSocket, TypeScript, React, Redux, Prompt engineering

Live Events Ticketing Platform

Project Description: This project is a mature music and live events platform focused on ticket sales, operating in long-term production and built around a modular monolith architecture with a shared database.

The system was actively evolving and required modernization of a legacy codebase while preserving backward compatibility and production stability. A key architectural challenge was extending a single Django-based backend to support multiple independent frontend applications backed by the same database.

I led the backend adaptation for this multi-frontend architecture by introducing an abstraction layer over the ORM, enabling source-aware data handling without duplicating business logic, while also contributing to frontend and administrative interface development to ensure consistent system behavior.

Domain: E-commerce, Live Events, Ticketing

Involvement Duration: 1 year

Project Role: Full Stack Developer

Responsibilities:

- Owned frontend development for user-facing applications and the administrative panel, including feature development and ongoing maintenance.
- Led backend modernization efforts for a mature legacy system built on Python 2.7 and Django 1.x, focusing on safe extensibility and backward compatibility.
- Designed and implemented backend support for multiple independent frontend applications sharing a single backend and database
- Built an abstraction layer on top of the Django ORM to enable source-aware data access and behavior depending on the requesting frontend.
- Solved architectural challenges related to data separation, request context propagation, and consistent business logic execution across multiple sites.
- Maintained and extended GraphQL APIs (Graphene-Django) to support new frontend requirements without breaking existing consumers.
- Ensured data integrity, consistency, and correctness across multiple client applications operating on shared models.

Project Team Size: 5 - 10 team members

Tools & Technologies: Python 2.7, Django 1, Django ORM, Graphene-Django, Docker, Docker Compose, GQL, Next.js 15, Apollo

Education	**Bachelor's degree** National Aerospace University -'Kharkiv Aviation Institute'
	Master's degree Aircraft Designing

Languages	**English –** Advanced **Ukrainian –** Native **Russian –** Native