

Misha Shpilevsky

Backend Engineer | Python | Fintech | AI

mshpilevsky@proton.me | Telegram: @MShpilevsky | github.com/Nicheye | youtube.com/@Codium123

PROFESSIONAL SUMMARY

Backend Engineer with 3+ years of experience building high-load fintech and SaaS products. Designed and delivered a crypto payment platform processing approximately 1,500 RPS and supporting \$100K in daily transaction volume. Deep expertise in Django/DRF, distributed systems, PostgreSQL performance, observability, and AI-enabled data workflows. Fluent in English.

CORE SKILLS

Python, Django, Django REST Framework, PostgreSQL, PgBouncer, Redis, Celery, Kafka, Docker, REST APIs, API Design, OAuth 2.0 / PKCE, Model Context Protocol (MCP), Microservices, Distributed Systems, CI/CD, Pytest, Grafana, Loki, Promtail, OpenAI API, Qdrant, Vector Search, Claude Code, Codex, Leadership, Project Management, Selenium, Git

AI & LLM EXPERIENCE

Production AI Engineer and part-time Head of AI at Timestripe, building AI-powered product capabilities that connect LLMs with real user workflows. Own architecture and production delivery of OpenAI-powered features, RAG systems, memory tooling, Qdrant vector search, and MCP integrations secured with OAuth/PKCE and resource-token enforcement.

Built an AI learning pipeline that turns textbook materials into structured lessons, combining LLM content generation with speech synthesis and speech recognition. Implemented pronunciation assessment by comparing user audio with reference pronunciation, with a focus on reliable asynchronous processing, clear API boundaries, secure authentication, and maintainable data models.

PROFESSIONAL EXPERIENCE

Timestripe | Software Engineer

2024 - Present

- Own product features end to end, from system design and implementation through production rollout and iteration; built collaboration, Spaces, onboarding, notifications, background jobs, recurring goals, public content, and timezone-aware scheduling.
- Designed APIs and integrations for Google Calendar, iOS, Trello, and Notion; took end-to-end ownership of the production MCP integration, including OAuth/PKCE, resource-token enforcement, and APIs for goals, time blocking, boards, comments, subgoals, sharing, folders, and users.
- Improved return conversion by 30% with a push-notification system; optimized database queries by more than 50% and increased SEO score by 20% with a customizable SEO-page framework.
- Focus on reliable asynchronous processing, secure authentication, clear API boundaries, and maintainable data models; integrated the OpenAI API for AI-powered product workflows.

Environment: Django REST Framework, Pytest, Celery, Redis, OpenAI API, OAuth 2.0 / PKCE, MCP, Git

Crypto Payment System | Backend Engineer

09/2024 - 11/2025

- Built a high-load crypto payment system from the ground up, supporting approximately 1,500 RPS and growing daily transaction volume to \$100K.
- Helped evolve the platform from a monolith to a microservices architecture; used PgBouncer to distribute database connections under load.
- Designed a dispute engine that reduced refund losses and increased net revenue by 20%.
- Delivered 15 API integrations with external payment providers and implemented two-factor authentication.
- Set up CI/CD for staging and production, introduced code review practices, and expanded automated-test coverage to 90% of the codebase.
- Built a Telegram bot that automated support workflows and reduced support costs by 30%.
- Implemented observability with Grafana, Loki, and Promtail; developed AI data-processing pipelines, a localization publishing service, and a SendGrid email-campaign service.

Environment: Django REST Framework, PostgreSQL, PgBouncer, Celery, Redis, Kafka, Qdrant, Grafana, Loki, Promtail, Docker, Pytest, CI/CD

Art. Lebedev Studio (Fokus) | Backend Engineer

01/2024 - 08/2024

- Developed an automated PDF-export system for working documents and presentations, improving generation speed by 115% and export capabilities by 70% through clickable HTML layouts.
- Built a dynamic redirect system for presentation landing pages, improving navigation and user engagement by 40%.
- Created a Telegram bot that identified the best presentations during each 24-hour period and automatically distributed email updates, improving communication efficiency by 70%.

Environment: Django REST Framework, Pytest, Selenium, Docker