

General

Name	Pavel
Education	GSU
Profession	Python engineer

Objective

Summary	Python engineer with more than 7.5 years of experience, adept at successfully completing a wide range of projects in terms of complexity and scale. I am well-versed in modern technologies and have a strong foundation in back-end development. I have led projects as a Team Lead, managing teams of 3-8 people, and have also worked as a DevOps engineer. My experience includes managing highly loaded systems and actively working with microservices. I have been involved in projects from the beginning, including building development environments and deploying servers. I have extensive experience in building application architecture, writing tests, and creating custom libraries. In addition, I have hands-on experience with AWS and have studied areas such as ML, TypeScript, and working as a fullstacker with React. I also have experience on the AI-project with GPT API, Phidata in my toolbox. I am willing to participate in the full development cycle, from implementing new features to improving existing features, writing technical documentation and conducting code reviews. My key skills include a strong command of Python with Django, Flask and FastAPI frameworks, as well as a deep understanding of basic algorithms and data structures. I also know OOP principles and design patterns and have experience with databases such as PostgreSQL.
Languages	English - B2
Industries	IT, HR, Business, Ecology, Fintech

Skills

Skill	Experience, years	Last used, year
MS Windows	7	2025
Linux (Ubuntu, Ubuntu Server)	7	2025
Git	7	2025
Python	7.5	2025
Django REST Framework	6	2025
FastAPI	4	2024
Flask	2	2024
React	2	2025
Docker (-compose)	5	2025
AWS (Lambda, S3, ECR, Cognito, API Gateway, CodeBuild, AWS AppSync, SM, CW)	3.5	2025
Azure	1	2019
PostgreSQL	5	2025
Elasticsearch	1	2022
MongoDB	2	2022
Redis	2	2024
SQLAlchemy	2	2024
Pydantic	1	2022
Celery	2	2024
HTML	5	2025
REST API	5	2025
GPT API	1	2024
Phidata	1	2024
CI/CD	2	2025
Aiohttp	3	2024

Requests	4	2025
Asyncio	2	2024
OAuth2	2	2025
Kafka	1	2021
Nginx	3	2025
Gunicorn	1	2023
WebSocket	2	2025
Github Copilot Agent	1	2025

Projects

Project 1	
Starts	08-2025
Ends	Till now
Description	A modern web-based educational platform designed to personalize the learning experience for middle and high school students. The foundation of the platform is the UK school curriculum, which has been digitized and enriched with artificial intelligence to generate interactive lessons and assignments. Each lesson is dynamically generated based on the official curriculum and enhanced with AI-driven content. At the end of each lesson, students take quizzes; the platform automatically grades results, analyzes mistakes, and provides personalized recommendations. The platform continuously tracks student progress and adjusts the learning path to strengthen weak areas while reinforcing strong skills.
Stack	Python, FastAPI, PostgreSQL, Docker (-compose), AWS(S3, ECS, CW), GitHub, GitHub actions
Role	Backend Developer
Achievements	- Designed and implemented project architecture from scratch, ensuring modularity, scalability, and maintainability. - Built authentication & authorization system using JWT with robust data validation (Pydantic v2) and unified API error handling. - Optimized database interactions with async SQLAlchemy sessions and query tuning, reducing API response times by 30% under production load. - Automated CI/CD pipelines (GitHub Actions / GitLab CI) with testing, linting, and zero-downtime deployments. - Containerized and deployed services in Docker & Kubernetes. - Implemented observability stack (Prometheus + Grafana) for proactive monitoring, alerting, and debugging in production.

Project 2	
Starts	08-2024
Ends	08-2025
Description	A fintech Django REST Framework web service designed with a monolithic architecture. The main functionality is to store and manage bank detail information. The system supported secure financial transactions and compliance with regulatory requirements. It included user roles and permissions management, fraud detection algorithms, and real-time transaction monitoring.
Stack	Python, Django REST Framework, FastAPI, Nginx, Docker (-compose), React
Role	Fullstack Developer
Achievements	• Optimized SQL queries for better performance. • Improved API endpoints for enhanced scalability. • Implemented pytest for application testing. • Configured and optimized Nginx infrastructure. • Developed fraud detection mechanisms to ensure transaction security. • Refactored codebase and architecture. • Communicated with a customer.

Project 3	
Starts	05-2024
Ends	07-2024
Description	The Telegram bot that communicated with a Telegram channel. The bot generated an HTML conversation and sent it to the Telegram channel when a user wrote in the bot and an answer wasn't found in a local knowledge base. In such cases, the bot sent a request to AI. The bot also provided automated responses for frequently asked questions, improving customer support efficiency. The system included logging and monitoring features for tracking user interactions and system performance.
Stack	Python, aiogram, PostgreSQL, Docker (-compose)
Role	TeamLead, Backend developer, DevOps
Achievements	● Optimized database queries for improved performance. ● Enhanced API endpoints for efficiency and response time. ● Implemented comprehensive testing using pytest. ● Improved bot response time by optimizing request handling. ● Developed a robust logging system for monitoring bot interactions. ● Communicated with a customer.

Project 4	
Starts	02-2024
Ends	04-2024
Description	The Telegram bot integrated as an API with the customer site and sent information from the site to the Telegram channel. The bot was designed to handle real-time notifications and updates from the customer's platform, ensuring seamless communication between the website and Telegram users. It also included admin functionalities for managing automated messages and monitoring system health.
Stack	Python, aiogram, PostgreSQL, Docker (-compose)
Role	TeamLead, Backend developer, DevOps
Achievements	● Optimized database queries for better performance. ● Refactored API endpoints to enhance efficiency. ● Implemented unit and integration testing using pytest. ● Developed an admin dashboard for tracking bot performance and logs. ● Communicated with a customer.

Project 5	
Starts	10-2023
Ends	01-2024
Description	The project is a FastAPI web service that processes large pharmacy data and determines the best way to buy or sell pharmaceutical elements such as pills. The main data processing functionality was implemented in AWS Lambda functions. The system also supported real-time analytics and price tracking to ensure customers received the best deals. A dedicated module was developed to manage inventory and predict demand trends based on historical sales data.
Stack	Python, FastAPI, PostgreSQL, AWS (lambda), Nginx, Docker (-compose)
Role	TeamLead, Backend developer, DevOps
Achievements	● Optimized database queries for faster data retrieval. ● Enhanced API endpoints to improve response time. ● Performed application testing using pytest. ● Configured infrastructure, including Nginx setup and optimization. ● Developed a scalable system for handling high traffic loads efficiently. ● Communicated with a customer.

Project 6	
Starts	02-2023
Ends	02-2024
Description	AI project. The project is a chatbot that supports multi-session interactions with users. Each user can have an individual conversation with the bot, and the chat history is preserved within their session. The chatbot functions similarly to OpenAI's ChatGPT, providing answers to questions, assistance, and other features based on its settings.
Stack	Python, Postgres, GPT API, AI Function calling, Phidata
Role	Backend Developer
Achievements	● Backend customization was performed for Chat GPT integration ● Backend and frontend connection ● Analyzing OpenAI documentation ● Connection of necessary endpoints ● Communicated with a customer.

Project 7	
Starts	04-2022
Ends	01-2023
Description	The project is a Django Rest Framework (DRF) web service designed with a microservices architecture. The main functionality of the project is to facilitate chat interactions via web sockets, integrating with Telegram and Viber bots. Additionally, the project includes a microservice written in FastAPI to handle sending messages via email. Overall, the project combines various technologies and architectural principles to create a robust and scalable system for facilitating chat interactions across multiple platforms, including Telegram, Viber, and email. It leverages the strengths of Django, DRF, FastAPI, and web sockets to provide a seamless user experience and efficient communication channels.
Stack	Python 3.9, FastAPI, DRF, PostgreSQL, WebSocket, Linux (VPS), Docker, Docker-compose, Kafka, Aiogram
Role	Team lead, Backend developer, DevOps
Achievements	• Implementation of a microservices architecture using Django REST Framework for a complex chat application. • Integration of multiple communication channels (Telegram, Viber) into a unified platform. • Efficient and scalable real-time communication through WebSocket implementation. • API design using RESTful principles and DRF features for optimal performance and maintainability. • Seamless integration of a FastAPI microservice for email handling, demonstrating effective polyglot microservices architecture. • Implementation of robust error handling and exception management to ensure system reliability. • Implementation of security measures to protect user data and prevent unauthorized access. • Project architecture development. • Setup environment on VPS, writing deploy scripts. • Team management (8 people). • Communicated with a customer.

Project 8	
Starts	07-2021
Ends	03-2022
Description	The project is a Python script that runs web microservices, each tasked with checking the contents of directories and their history. The microservices are initiated using an XML configuration file, which dictates their behavior and parameters. The project provides a flexible and customizable solution for monitoring directory contents and their history using web microservices. By leveraging Python for scripting and web development, it offers a lightweight and efficient way to manage and track changes in directories according to user-defined configurations.
Stack	Python 3.9, FastAPI, Linux, XML, RabbitMQ, Gunicorn, SQLAlchemy, Redis, OAuth2.0
Role	Team lead, Backend developer
Achievements	• Development of a flexible and scalable microservices architecture using Python and FastAPI for directory monitoring. • Implementation of a configuration-driven system using XML for dynamic microservice behavior. • Efficient directory and history monitoring through optimized Python scripts. • Implementation of a public async API for external directory and history checks. • Code refactoring and optimization for improved performance, maintainability, and bug reduction. • Effective communication and coordination with stakeholders and customers and suggestions to optimize code. • Proactive identification and resolution of project challenges. • Writing tests. • Communicated with a customer.

Project 9	
Starts	03-2021
Ends	06-2021
Description	The project is a web service designed specifically for testing and processing candidate statistics. It provides a platform for assessing candidates' performance, gathering relevant data, and generating insightful analytics to aid in decision-making processes. The project aims to streamline the testing and assessment process for candidates while providing administrators with comprehensive insights into candidate performance. By leveraging web technologies and statistical analysis, it offers a powerful tool for optimizing recruitment and evaluation processes in various domains.
Stack	Python 3.9, DRF, PostgreSQL, Linux, Nginx, AWS, Redis
Role	Backend developer
Achievements	• Developed a robust candidate testing and analytics platform using Python, DRF, and PostgreSQL. • Created a public API for efficient test administration and data collection. • Optimized data processing pipelines resulting in a significant performance increase (up to 3x). • Effective DRF using for building a scalable and maintainable API. • Integrated PostgreSQL for reliable and efficient data storage and retrieval. • Implemented robust error handling and data validation to ensure data integrity. • Streamlined candidate testing process through user-friendly API. • Comprehensive candidate data collection and analysis. • Communicated with customers and suggestions to optimize code.

Project 10	
Starts	04-2020
Ends	02-2021
Description	Climate data analytics, ESG metrics, and predictive insights on carbon exposure & energy transition.
Stack	Python 3.8, FastAPI, AWS (lambda, S3, ECR, Cognito, API Gateway, CodeBuild, AWS AppSync, SM, CW), Azure, TypeScript
Role	Backend developer
Achievements	• Development of a robust climate data analytics platform leveraging Python, FastAPI, and AWS. • Creation of a public API for collecting and processing environmental emissions data. • Integration of AWS AppSync for GraphQL synchronization. • Implementation of data processing pipelines for extracting insights from environmental data. • Added the ability to process statistics in the background using data from S3 and AWS lambda. • Communication with customers and suggestions to optimize code and use AWS resources to reduce cost.

Project 11	
Starts	12-2019
Ends	03-2020
Description	The project is an Advertisement Management System designed to streamline the process of creating, managing, and analyzing advertising campaigns across various sites.
Stack	Python 3.8, FastAPI, AWS (lambda, S3, Amazon Kinesis)
Role	Backend developer
Achievements	• Development of a scalable and efficient advertisement management system using Python, FastAPI, and AWS. • Creation of a robust public API for data collection and processing related to ad displays. • Effective utilization of AWS Lambda and Kinesis for real-time data processing and handling. • Optimized codebase for improved performance and scalability. • Communicating with the customer.

Project 12	
Starts	03-2019
Ends	11-2019
Description	Biology service: advice and support. The service can provide advice and support in the field of molecular biology, including the development of experimental designs and interpretation of results.
Stack	Python 3.6, Flask, FastAPI, MongoDB, Celery, AWS localstack, Marshmallow, Elasticsearch
Role	Backend developer, Team lead
Achievements	• Development of a robust biology advice and support platform using Python, Flask, FastAPI, MongoDB and Elasticsearch. • Implementation of a public API to provide data for other services, enhancing system interoperability. • Optimization of existing API endpoints for improved performance and efficiency. • Creation of a comprehensive unit and integration test suite to ensure code quality and reliability. • Significant improvement in test execution speed through test redesign. • Identification of opportunities for further optimization and UX improvements.

Project 13	
Starts	11-2018
Ends	02-2019
Description	The project is an office management system designed to automate the allocation of workspace for employees within a building layout. It facilitates the efficient distribution of employees across available workstations based on various criteria and requirements. The project aims to streamline the management of office space and resources by automating the allocation of workspaces for employees. By leveraging data-driven algorithms and visualization tools, it provides administrators with the tools they need to optimize workspace utilization and create a productive and efficient work environment.
Stack	Python 3.8, Django REST Framework, PostgreSQL, Asyncio, Aiohttp
Role	Backend developer, Team lead
Achievements	• Development of a robust office workspace allocation system using Python, Django REST Framework, and PostgreSQL. • Creation of a public API for seamless integration with the company's general system. • Implementation of efficient workspace allocation algorithms to optimize space utilization. • Integration with external systems like BambooHR to enhance data accuracy and consistency. • Development of a comprehensive test suite to ensure system reliability and accuracy.

Project 14	
Starts	08-2018
Ends	10-2018
Description	The project is a platform to choose users by photo and communicate with users liked by photos.
Stack	Python, Django REST Framework, Celery, PostgreSQL
Role	Backend developer
Achievements	• Development of a platform based on Python, DRF, and FastAPI. • Creation of a robust public API for integration with various services and systems. • Effective use of Celery for asynchronous task handling. • Optimized codebase through refactoring for improved performance and maintainability. • Refactoring code. • Communicating with the customer.

Project 15	
Starts	05-2018
Ends	07-2018
Description	The project is a trading platform that leverages a microservices architecture to provide a scalable, modular, and flexible system for conducting financial transactions and managing trading activities.
Stack	Python, Django REST Framework, FastAPI, Celery, PostgreSQL, Kafka
Role	Backend developer
Achievements	• Development of a scalable and flexible trading platform utilizing a microservices architecture based on Python, DRF, and FastAPI. • Creation of a robust public API for integration with various services and systems. • Implementation of efficient microservices for stats collection, processing, and email notifications. • Effective use of Celery for asynchronous task handling. • Optimized codebase through refactoring for improved performance and maintainability. • Refactoring code. • Communicating with the customer.

Internship	
Starts	07-2017
Ends	04-2018
Description	The ML and BigData internship in IBA group. The tasks were develop tasks with PySpark and big files, weather forecast, MNIST.
Stack	Python, ML, scypy, numpy, pandas, PySpark, PyTorch
Role	Trainee
Achievements	• Implementation an own spark cluster • Developing multiple tasks working with big data • Developing task with weather forecast • Developing a base MNIST taks