# Mohamad Liyaghi *Python Back-End Developer*


mohamad-liyaghi.me

liaghimohamad69@gmail.com

mohamad-liyaghi

Tehran, Iran

+98 991 951 4231

mohamad-liyaghi


## Profile

Passionate back-end developer with 2 years of experience building RESTful APIs and web services using Django, Django REST Framework, and FastAPI. Completed over 10 successful freelance projects and contributed to over 18 open-source projects. Able to lead small teams of developers and continuously learn and apply new technologies and industry best practices. Committed to adhering to best practices, staying up-to-date with the latest technologies and trends.

## Skills

Django • Fast Api • Docker • Celery • Redis • Sql • Mongo DB • Nginx

## Projects

### Academy Master

An open-source web application built with Django that streamlines academic institution management. It leverages Django's built-in permission system to provide different access levels for administrators, managers, teachers, and students. The application also integrates technologies like Elastic Search, Docker, Celery, and Redis to enhance its functionality.

### Tsuna Streaming

Advanced streaming API system allows users to create channels, add administrators, and subscribe to channels. Users can add video/music to their channels, and other users can vote and comment on the content. The system is built using Django Rest Framework (DRF) and utilizes Redis for caching, Nginx for web serving, Pytest for testing, and Celery for background tasks.

### Fast Commerce

An e-commerce API using FastAPI and SQLAlchemy that enables users to become vendors and sell products. Vendors can add products with details. Customers can browse and search products, add items to a cart, and create orders. Used Redis for storing cart data. Implemented Docker containerization and CI/CD pipelines for efficient deployment. Tested API using Pytest.

### Anon Market

An e-commerce project that allows users to trade anonymously using a custom website currency. In addition to facilitating secure transactions, the platform features real-time chat functionality and a blog and forum system. The project uses Django and utilizes PostgreSQL as the database, Nginx as the web server, Docker for containerization, and Django Channels for real-time communication. The resulting platform provides users a seamless and engaging experience, while prioritizing security and privacy.

## Languages

**Persian** (Native) • **English** (B2) • **German** (A1)