



# SHAHAB HAMIDI

## BACK-END DEVELOPER

### CONTACT

 PHONE  
+98 919 414 2303

 EMAIL  
shahabhamidi88@gmail.com

 GITHUB  
[github.com/shahabhamidi](https://github.com/shahabhamidi)

 LINKEDIN  
[linkedin.com/in/shahabhamidi](https://linkedin.com/in/shahabhamidi)

LOCATION  
Tehran, Iran

### SKILLS

#### Languages & Frameworks

Python · Django · Django REST Framework · REST APIs

#### Databases

PostgreSQL · MySQL · SQLite

#### DevOps & Infrastructure

Docker · Linux · Git · Redis · Celery · CI/CD · Kubernetes

#### Architecture & Practices

SOLID Principles · System Design · Design Patterns

#### Networking & Security

TCP/IP · HTTP · DNS · OWASP Top 10 · CSRF & SQL-Injection Prevention · Password Hashing

### CERTIFICATIONS

#### Python Programming

#### Django Framework

#### Task-Oriented Course to Security

### EDUCATION

#### B.Sc. Software Engineering

Kharazmi University, Tehran

#### National Organization for Development of Exceptional Talents

(SAMPAD)

### INTERESTS

Back-End Development · Cybersecurity · Open Source · Problem Solving · Linux

### PROFILE

Backend developer working with Python and Django, focused on building REST APIs and services that hold up in production. Comfortable across the backend lifecycle — API design (DRF), relational and caching layers (PostgreSQL, Redis), background processing (Celery), and containerized delivery (Docker, Kubernetes, CI/CD). Grounded in SOLID principles, system design, and design patterns, with a security-conscious approach to web development.

### EXPERIENCE

#### Backend Developer

IPCO · Present

- Build and maintain backend services and REST APIs with Django and Django REST Framework.
- Work with PostgreSQL and Redis for data storage and caching, and Celery for background task processing.
- Contribute to containerized deployments and CI/CD pipelines.

### PROJECT EXPERIENCE

#### Django Multi-Shop

Multi-vendor e-commerce platform · Django (MVT), Django ORM, MySQL, HTML/CSS/JS

- Designed and built a multi-vendor architecture where each vendor manages an independent storefront — products, inventory, and orders — on a shared codebase.
- Implemented role-based authentication separating customer, vendor, and admin permissions using Django's auth framework.
- Built full product management (CRUD, categories, inventory, images) and a session-based cart-to-checkout flow that generates orders and invoices per vendor.
- Developed a vendor dashboard for managing products, orders, and stock, plus a lightweight analytics view summarizing sales and performance.

#### Technical challenges solved:

- Isolated vendor data at the ORM layer so each storefront only ever sees and modifies its own records.
- Modeled the many-to-many relationships a multi-vendor marketplace needs between vendors, products, and orders.
- Enforced a role-based permission system on top of Django auth to keep vendor and admin actions properly scoped.

### WHAT I BRING

- A builder's mindset — I ship complete, working systems rather than isolated snippets, and enjoy owning a feature from schema to UI.
- Solid grounding in SOLID principles, system design, and design patterns for backend architecture that scales.
- A security-conscious approach to backend work, from hashing and CSRF protection to SQL-injection prevention.
- Comfort with the full delivery pipeline: Git, Docker, Kubernetes, and CI/CD, on top of daily Linux CLI use.