VLADISLAV CHERKASHYN

PYTHON DEVELOPER

CONTACTS

+ 38 (095) 418 68 92
dashevskiy.cherkashin.vladislav
@gmail.com
Telegram / GitHub / LinkedIn
Vinnytsia, Ukraine
Remote / office

SKILLS

- **Python** 3.7+ & **OOP**
- Design Patterns, TDD, SOLID, DRY, KISS
- Algorithms, Data Structures
- Web Frameworks: **Django**, **DRF**, Flask (basics), FastAPI (basics)
- Databases: **PostgreSQL**, SQLite, Redis (basics), MongoDB (basics)
- Multiprocessing
- Front-end: HTML, CSS, Bootstrap, JavaScript (basics)
- DevOps: **Docker**, Render.com
- Networking & Security: HTTP, SSL/TLS, JWT
- Tools: Git, GitHub, Debugger, Pytest, Trello
- English: **Upper-Intermediate**

EDUCATION

MASTER'S DIPLOMA
WITH HONORS IN
HIGH ENERGY PHYSICS
AND NUCLEAR PHYSICS
V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY 2018 - 2020

Python developer with skills in Django and DRF. My technical education has enabled me to successfully tackle technical challenges, and my experience in QA engineering gives me an understanding of the importance of quality and reliability of the product, which are key factors in successful back-end development. I am committed to overcoming challenges and pushing through obstacles.

WORK EXPERIENCE

PYTHON DEVELOPER

MY PROJECTS| FEB 2023 - NOW

- Django-based task manager (Python, Django, deployed to Render)
- Restful API to manage book borrowing system (Team Project)
(Implementation models, payment method via Stripe, Docker, Swagger)
- Binance data collector (Python, Binance API, Flask, Docker, PostgreSQL, plottly)

QA ENGINEER

YELDLY | MAR 2021 - AUG 2022

- Testing documentation maintenance.
- Manual tests covering the entire front-end part of the project
- Verification of transactions compliance

CUSTOMER SUPPORT SPECIALIST

ZONE3000 | AUG 2020 - FEB 2021

- Responding to incoming messages and questions
- Customization of the company's products for convenient use by customers
- Working with negative feedback, forming brand loyalty

BRIEF DESCRIPTION OF RESPONSIBILITIES AND TECHNOLOGIES OF THE PROJECTS

- Django-based task manager
  Technologies: Python, Django, PostgreSQL, HTML/CSS, Bootstrap, GIT, API.

The project was developed using the Django framework with the standard MVT (Model-View-Template) architecture. The following features were implemented:

1. The logic for adding, deleting employees and tasks, editing their information, associating tasks with employees, marking completed tasks, and performing database-wide searches using Django ORM. PostgreSQL was used as the storage solution to ensure data integrity.
2. The task table highlights overdue tasks, it is possible to mark the task as completed in the general task table
3. Pagination of pages was implemented to handle large datasets and improve user experience.
4. User registration (login and logout) functionality was implemented using built-in Django framework tools and customized for the task manager application.
5. HTML, CSS, and Bootstrap were used for building the user interface and styling pages. JavaScript and jQuery were also incorporated to provide dynamic functionality and enhance the user experience.
6. CSRF token protection was implemented for all views with the POST method to prevent cross-site request forgery attacks.
7. The application was deployed on Render.com to make it accessible online.

- Restful API to manage book borrowing system (Team Project)
  Technologies: DRF, PostgreSQL, Celery, Redis, Stripe API, Telegram API, Swagger, Docker)

Responsibilities:

1. Developed models for the book-borrowing functionality of the application.
2. Implemented Stripe payment method integration.
3. Created documentation using the Swagger toolset.
4. Ensured test coverage for the application.
5. Containerized the application using Docker and connected it to PostgreSQL.

- Binance data collector
  Technologies: Python, Binance API, Flask, Docker, PostgreSQL, plottly

1. Implemented logic in Flask framework to fetch data from Binance API for selected trading pairs and time intervals.
2. Utilized Plotly library for visualizing ratio changes for chosen pairs and intervals.
3. Developed functionality to save data to both CSV format and PostgreSQL database concurrently using threading library.
4. Containerized the application using Docker.