

Baranov Timofei

Call me: +373 775-977-07

E-mail me: timofeib11@gmail.com

Website: timofeib.info

Address: Vilnius, Lithuania

LinkedIn: [LinkedIn](#)

Available Upon Request

GOAL

Aspiring to become a Machine Learning Engineer in a team focused on developing and improving different aspects in AI and machine learning. I am eager to leverage my strong skills and experience in building and managing complex infrastructure supporting MLOps, cloud technologies, and deployment systems. Experienced in decentralized and Web3 environments, I aim to contribute to the advancement of new technologies and innovative solutions that enhance system efficiency and security.

EDUCATION

- Humanities and Mathematics Gymnasium of the city of Tiraspol **2014-2022**
 - **AVG:** 5.00/5.00
- Theoretical General Education Lyceum of Tiraspol **2022-2023**
 - **AVG:** 4.88/5.00
- Vancouver Island University High School of Nanaimo **2023-2024**
 - **GPA:** 3.1/4.00
- Aspengrove School of Nanaimo - The International Baccalaureate **2024-2026**
 - **IB:** 32/42
- Vilnius Tech University **2026-20..**

HIGHLIGHTS OF QUALIFICATIONS

- Proficient in version control systems, particularly **Git**.
- Strong experience in **Python**, with extensive use of the **Django** framework for web backend development.
- Basic experience with **JavaScript**, **C#**, **SQL** (PostgreSQL), and **Pascal**.
- Familiar with **HTML** and **CSS**, including Bootstrap.
- Solid understanding of **Object-Oriented Programming** (OOP) and **API** integration.
- Experienced in using **DBeaver** for database management.
- Great experience in using Claude for complex tasks with a strong knowledge of good prompting.
- Knowledgeable in dropshipping and proficient in using the **Shopify** platform.
- Basic experience in crypto-trading.
- Proficient in Microsoft Office applications.

FINISHED PROJECTS

- **Personal website** built using Python, HTML, CSS, JavaScript.
- **Sepira** - a website designed to manage a system of goods. This platform facilitates the allocation of goods into specific sections within a warehouse using only their QR codes.
- **Chess Board Detection System** is a computer vision project designed to detect and analyze a chessboard using live video from a camera.
- **AI Model for Clothing Recognition** - PyTorch-based image classification project built on the FashionMNIST dataset.
- **AI 101 Academy** - An interactive AI education platform. Django + PostgreSQL backend, Tailwind-styled frontend with React components, Stripe payments, PDF certificates, and a proactive AI tutor.
- **Atmos** - Atmos is an Android app that dynamically changes the device wallpaper based on current weather conditions. It adapts the wallpaper to reflect the weather, creating a more immersive and personalized experience.
- **Crypto Bot** (Currently working on it) - Crypto Trading Bot that uses 5 agents: agents 1 - 4 turn Binance candles, news, and order-flow data into pure, untrained numeric features (patterns, indicators, sentiment, tape). Agent 5 is the only AI model that uses scikit-learn logistic + isotonic calibration and optional LightGBM to combine the data from the first 4 agents and predict the outcome.

FINISHED COURSES

- Python
- Git - version control version
- HTML & CSS for the beginners
- Robotics
- Excel - basics
- OOP and working with API
- Claude Code in Action

Netology

Netology

College's course

University's program

Netology

Netology

Anthropic

OTHER ACTIVITIES & INTERESTS

Activities:

- VEX V5 Robotics
- Professional Tennis Player
- Active Volleyball Player
- BMX Rider

Languages:

- English – **C1** (IELTS Verified)
- Russian – **Native**

Personal Projects Portfolio

Timofei Baranov

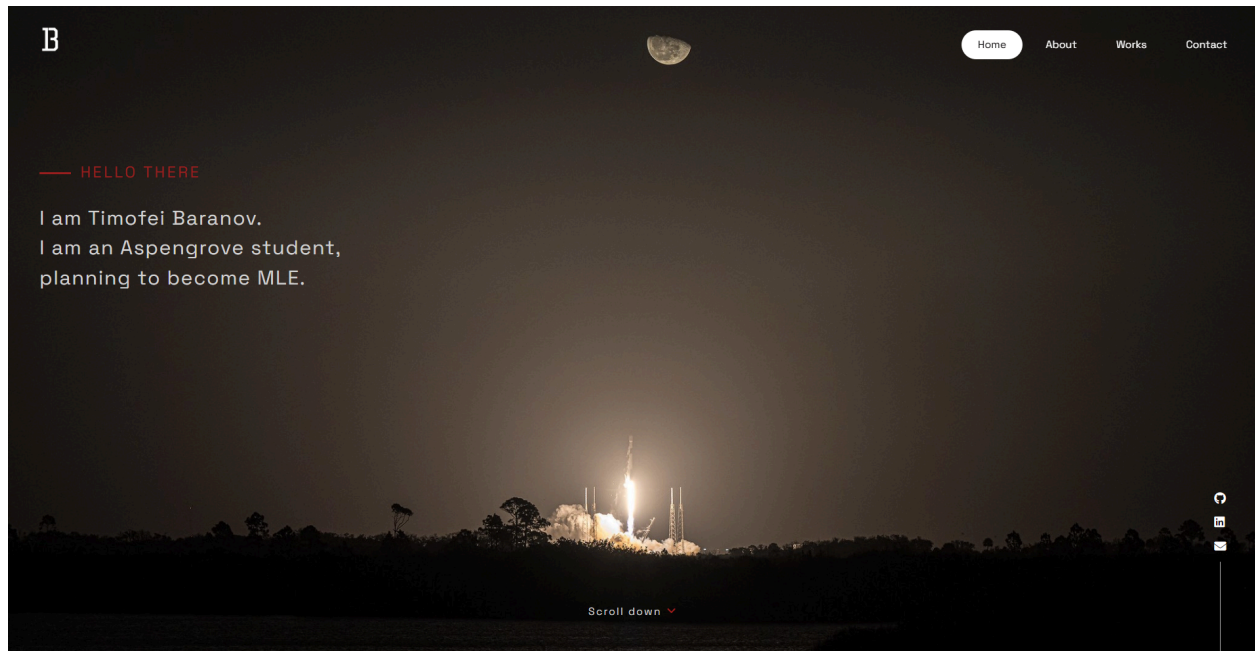
LinkedIn: [Link](#)

For paper copies: <https://linkedin.com/in/tim-baranov>

GitHub: [Link](#)

For paper copies: <https://github.com/BaranovTim>

Personal Website



Description

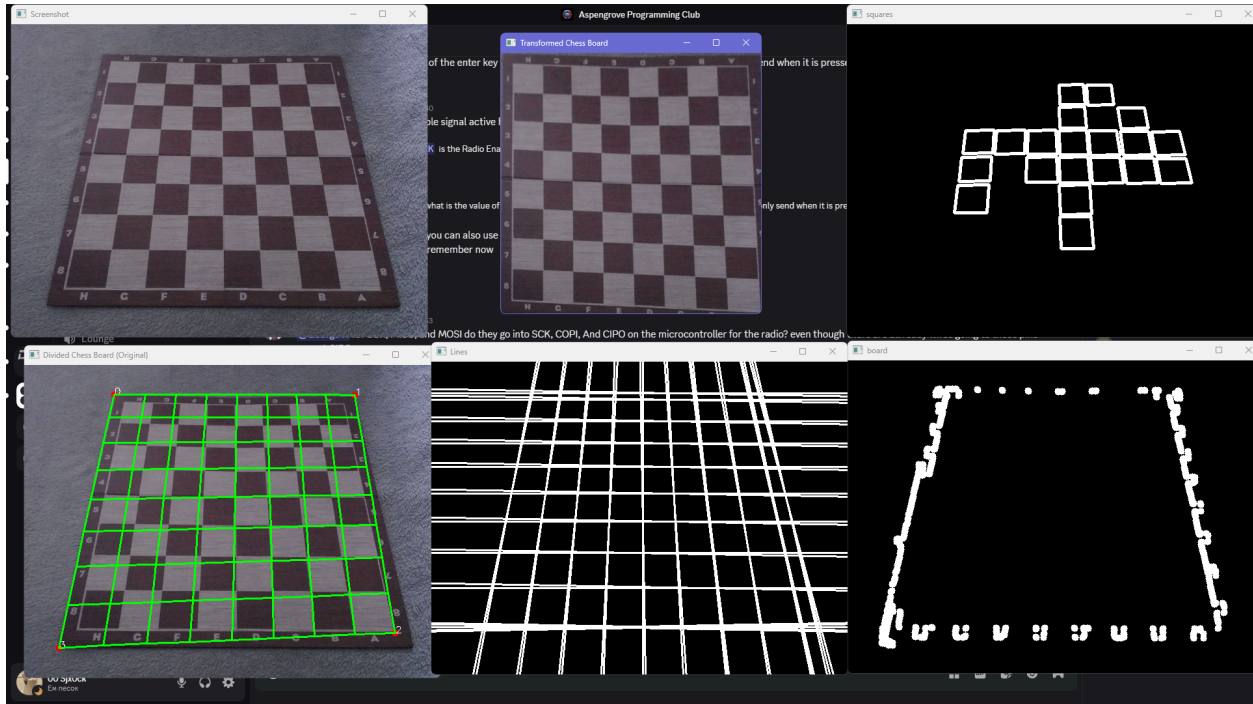
timofeib.info is a personal portfolio website built to present information about me, my projects, and my technical skills. The website was developed using Python with Django for the backend, and HTML, CSS, and JavaScript for the frontend. It is deployed and running on Vercel.com.

The website is fully responsive and works smoothly on all screen types, including TVs, phones, PCs, and tablets. It includes sections about me, my programming projects, the coding languages and technologies I know, and other interactive and creative elements that showcase my interests in software development.

The goal of this project was to create a modern, fast, and accessible platform that clearly represents my skills, experience, and passion for technology.

[GitHub Repository](#)

Chess Board Detector



Description

Chess Board and Piece Detection System is a computer vision project designed to detect and analyze a chessboard using live video from a camera. The system processes frames in real time, identifies the board layout, and provides clear visual feedback including bounding boxes, labels, and coordinate data.

The program is written entirely in Python and uses the OpenCV library for image capture and processing. It allows users to view a live camera feed, detect the chessboard, and display detailed information such as square positions and grid overlays for analysis or integration into other chess-related applications.

Future Development: Planned integration with Meta AI Glasses for hands-free board tracking. Future improvements will include piece detection and automated move recording.

[GitHub Repository](#)

AI Model for Clothing Recognition



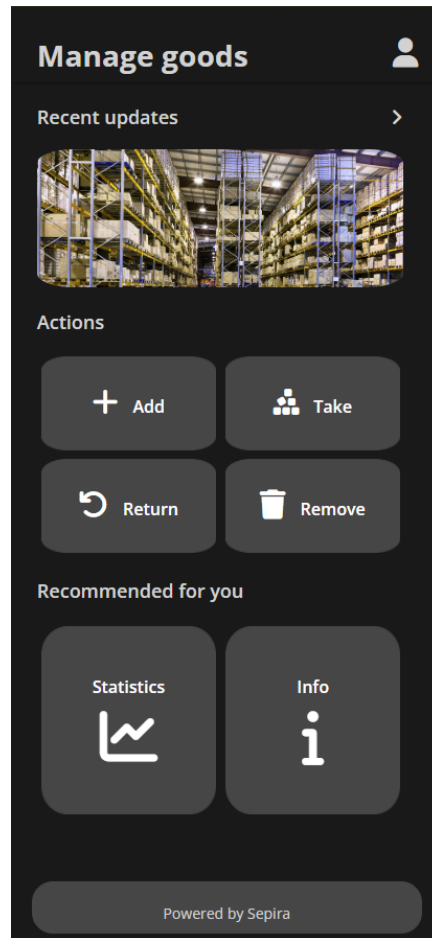
Description

Clothing Recognition is a PyTorch-based image classification project built on the FashionMNIST dataset. The repository contains a custom convolutional neural network implementation, training and evaluation loops, utilities for prediction on sample images, and visualization scripts for performance analysis (precision/recall curves, confusion matrix, F1 vs confidence).

Link to my Research project on this AI model is [here](#).

[GitHub Repository](#)

Sepira



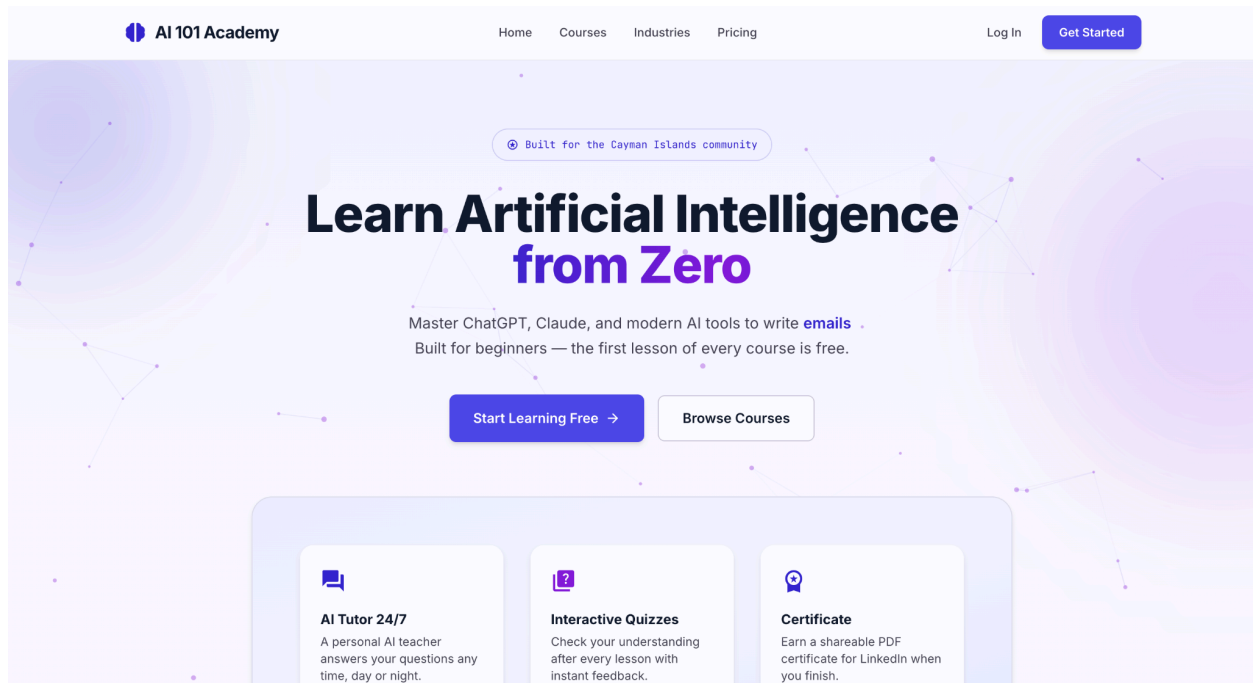
Description

Sepira is a website designed to manage a system of goods, allowing for their creation, modification, and organization. This platform facilitates the allocation of goods into specific sections within a warehouse using only their QR codes. Workers can add items to the warehouse, retrieve them, and monitor quantities. All operations are supported by a PostgreSQL database.

The QR code scanner is implemented in JavaScript, while the backend leverages the Django framework written in Python. Additionally, there are unique features for superusers - individuals responsible for managing the database. Superusers have access to advanced functionalities, such as adding news updates to ensure workers stay informed about the latest developments within the warehouse, as well as tracking worker activities, including the time, item, and quantity involved in each operation.

[GitHub Repository](#)

AI 101 Academy



Description

An interactive AI education platform. Django + PostgreSQL backend, Tailwind-styled frontend with React components, Stripe payments, PDF certificates, and a proactive AI tutor.

How it works:

- Learners buy tracks, not individual courses: the General track is the base purchase, and career tracks (Accounting, Hospitality, Banking, ...) can be added on top. One payment unlocks every course in the track — forever, including courses added later.
- Sign up and pick a career at signup — the matching career track is recommended first on the catalog, pricing page, and dashboard.
- The first lesson of every course is free (including its quiz and the AI tutor).
- Verified students (upload a student ID in Settings, staff approves) automatically get the track's student price.
- Progress tracking, interactive quizzes, a proactive AI assistant that pops up to help after wrong quiz answers, and a professional PDF certificate per completed course.

[GitHub Repository](#)

Atmos



Description

Atmos is an Android application that connects real-time weather conditions with the device's wallpaper, automatically adapting the visual background to match the current atmosphere. The app is built with Dart and uses Flutter for the application interface and Android functionality, combining weather data with automated wallpaper management to create a dynamic and personalized experience.

[GitHub Repository](#)