VINICIO ALEJANDRO NARANJO MOSQUERA

Phone number: +34 711717713

Email: vinicio.naranjo15@gmail.com

GitHub Repository: https://github.com/6aligula

PROFESSIONAL EXPERIENCE

2023-2024 Programming Teacher.

2022-2023 Freelance: Software Developer

- Use of Python in back-end application development and cybersecurity, combined with React Native on the front-end to create attractive and functional user interfaces.
- Experience in ChatGPT engineering for developers, specifically with GPT-4.
- Versatile web developer with skills in Django, React Native, C#, Entity Framework, Dapper, Windows server, JavaScript, Jquery, HTML, CSS, Bash and Aws.
- Familiar with AI tools like GPT-4 and GitHub Copilot for optimizing the development and deployment of AWS cloud applications.
- Experience in designing and implementing database management systems, with a deep understanding of object-oriented and functional programming principles.
- Proficiency in AWS services, including EC2, S3, and VPS.
- Over 10 years of experience in hardware reverse engineering, complemented by competencies in programming and desktop support.
- Record of successful projects and challenging problem-solving thanks to a combination of reverse engineering experience and programming skills.

2022-2022 Junior Programming

- Designing and development of an aplication for Stock electronic laboratory control. Using of programming languages: PHP, JavaScript; Jquery, AJAX, MySQL, Java.
- Designing of Data Base structure.
- Researching of new languages: Rust, Python, C, C++.

2017-2022 Electronic Technician in JZ Recielca

- Use of reverse engineering as a tool for the diagnostic and repairing of fork lift failures.
- Customer Service for whole diagnostic of failures.
- Home assistance for repairing by order of the customers.
- Diagnosis and repairing of speed controllers for DC motors and frequency inverters.
- Manufacturing of fork lift electrical wiring.

2016-2017 Electronic Technician in Tryo Aerospace

- Assembly of FM equipment (SMD manual welding) Microscope welding.
- Work under ESA Regulation guideslines:
  - ✓ ECSS-Q-ST-70-08C (THD Welding)
  - ✓ ECSS-Q-ST-70-38 (SMD Solder)
  - ✓ ECSS-Q-ST-10-09 (Warranty Non-Conformities)

2014-2016 Electronic Technician in HP Inc

- Designing of PCB with Cadence Software.
- Manufacturing of PCBs (LPKF Protomat S43 and Protomat S103 Milling Machine).
- PCB assembly with the required electronic devices.
- SMD welding with microscope.
- Testing and repairing of PCBs with automated Ersa Rework Hibrid.
- Repairing by reballing techniques of microcontrollers.
- Manufacturing and assembly of electrical circuits for new printer devices.
- Control and management of laboratory stock.
- Designing and manufacturing cables for electrical wiring printers. Industrial connectors used: DB9 family, Picospox, milligrid, spox, minifit JR, minifit Sr, microfit, sable, super sable, KK, Maten'look...)

ACADEMIC CAREER

2023	Course part of data science degree in introduction to artificial intelligence & Data Science degree(studying). Educational Center: UOC
2023	Master: Cyber Security. Educational Center: Unir.
2022	Bachelor technician: Development of multiplatform applications. Educational Center: Ins Calamot.
2013	Bachelor technician: Electronic Maintenance. Educational Center: Ins Camps Blancs

Languages

Spanish: Native

English: Technical

Catalan: C1

PROGRAMMING LANGUAGES

Bash, Python, JQuery, SQL, JavaScript, HTML, CSS, C, C++, C#, Java, Php, Bootstrap, Dart, Rust.

PERSONAL PROJECTS

Self-Built Smart Home System

- Designed and constructed a smart home system utilizing Raspberry Pi and ESP32 as primary controllers.
- Implemented MQTT communication to interconnect devices and enable efficient real-time communication between them.
- Developed a mobile application for remote control and monitoring of devices within the home.
- Established a VPN connection for secure remote access to the system, ensuring data protection and privacy.
- Integrated multiple sensors and actuators to perform specific functions.

eCommerce Platform on AWS

- Developed an eCommerce platform using Django as the backend, hosted on AWS (Amazon Web Services).
- Secured the entire platform, implementing best practices to protect user data and transactions.
- Constructed the frontend using React Native, enhanced with essential libraries such as Redux for state management, Secure Store for secure data storage, and AsyncStorage for asynchronous, unencrypted storage.
- Established HTTPS communication between frontend and backend, ensuring data encryption and secure data transfer.

Project Description: E-commerce Application with 4-Layer Architecture

Project: Development of an e-commerce application using .NET Core 3.1 with a 4-layer architecture for a specific client.

Project Overview:
The project involved developing a robust e-commerce application that provides a seamless user experience for both administrators and end-users. The application implements critical functionalities such as user management via Identity Framework, a shopping cart system for unauthenticated users, a checkout process with payment simulation using Iyzipay, and admin and user panels for managing and tracking products, orders, and campaigns.

Developed Functionalities:

User Registration and Management:
- **Technology**: ASP.NET Core Identity Framework.
- **Functionality**: Users can register, update, and modify their profiles. User registration includes email confirmation using Google SMTP Mail services.
- **Security**: Implementation of authentication and authorization to protect resources and sensitive data.

Admin Panel:
- **Product Management:** Create, update, and delete products in the store.
- **Order Management:** Monitor and control the status of orders.
- **Campaign Management:** Create and manage promotional campaigns.
- **Statistics**: View key statistics for decision-making.
- **Interface**: Implementation of an intuitive panel for managing all aspects of the e-commerce platform.

User Panel:
- **Order Tracking:** Allows users to track the status of their orders in real-time.
- **Profile Management:** Users can update their personal and contact information.
- **Interface**: A simple and accessible user panel focused on enhancing the customer experience.

Shopping Cart and Checkout Process:
- **Shopping Cart:** Implementation of a shopping cart accessible to unauthenticated users, with mandatory registration or authentication required before completing a purchase.
- **Checkout**: Simulation of credit/debit card payments through integration with Iyzipay, ensuring a smooth and secure purchase process.

Project Architecture:

1. Core (E-Commerce-App.Core):
- **Role**: Contains domain entities, service and repository interfaces, and shared business logic. This layer defines the fundamental structure of the project, ensuring that business rules are centralized and reusable.
- **Modularity**: Each entity and business logic component is separated to facilitate scalability and maintainability of the code.

2. Data (E-Commerce-App.Data):

- **Role**: Implements repositories, data access configuration via Entity Framework, and manages migrations.
- **Persistence**: Use of Entity Framework Core to efficiently handle database operations, including migrations and seeding of initial data.

3. Business (E-Commerce-App.Business):

- **Role**: Implements business logic that directly interacts with the data layer. This layer is responsible for applying specific business rules to the data managed within the application.
- **Services**: Implementation of services that encapsulate business logic, ensuring that operations are performed according to the client's requirements.

4. WebUI (E-Commerce-App.WebUI):

- **Role**: Contains the user interface, controllers, and views, managing the final user interaction with the application.
- **User Interface:** Development of views and controllers using ASP.NET Core MVC, with AutoMapper integration to simplify data transformation between the business layer and the user interface.
- **Security and Authentication:** Implementation of ASP.NET Identity with support for user authentication and resource authorization.

Architecture Benefits:

- **Modularity:** Clear separation between business logic, data management, and presentation, allowing for easy scalability and maintenance.
- **Reusability:** The Core and Data layers can be reused in future projects, reducing long-term development costs.
- **Maintainability:** The structured organization of code facilitates the implementation of new features and the resolution of bugs without compromising system stability.

Technologies Used:

- .NET Core 3.1
- Entity Framework Core 5.0
- ASP.NET Core Identity Framework
- AutoMapper
- Iyzipay (payment simulation)
- Google SMTP Mail (for email confirmation)

This project represents a comprehensive e-commerce system developed with best practices in architecture and a modular approach, ensuring quality and efficiency in both development and maintenance phases.

Project Description: Water Management Application with Dockerized Backend

Project Overview:
This project involved developing a comprehensive water management application tailored for communities or companies that require efficient water usage monitoring, user registration and management, invoice issuance, and circular distribution. The backend is implemented using Django, with Docker utilized for containerization to ensure scalability, security, and ease of deployment.

Key Features:

1. User Registration and Management:
- Implemented using Django and Django REST Framework, the system allows users to register, manage their profiles, and securely authenticate via JSON Web Tokens (JWT).
- Security is enhanced through JWT authentication, offering customizable token lifetimes and stateless session management.

2. Water Usage Monitoring and Control:
- Geospatial data is managed and queried using `django.contrib.gis` and `rest_framework_gis`, enabling precise monitoring of water resources.
- The system integrates with Leaflet through `django-leaflet` to provide interactive maps and visualizations, enhancing the user experience in monitoring water usage.

3. Invoice Generation:
- The system generates invoices based on water usage data, using ReportLab for PDF generation.
- Users can download invoices or receive them via email, ensuring transparency and accountability in billing.

4. Circulars and Communications:
- The system allows administrators to create and distribute circulars, facilitating effective communication within the community or company.
- Circulars are distributed via email, leveraging Django's email backend configuration.

Technical Architecture:

- Backend Framework: Django 4.0 with Django REST Framework.
- Database: PostgreSQL with PostGIS for geospatial data handling.
- Containerization: The backend is fully containerized using Docker, with Docker-Compose orchestrating the multi-container setup.
- Authentication: Implemented using `rest_framework_simplejwt` for JWT-based secure user sessions.
- Email Integration: Configured SMTP backend for sending invoices and circulars.
- Geospatial Capabilities: Powered by Django's GIS capabilities and PostGIS for precise geospatial data management.

Project Benefits:

- Scalability: Dockerization ensures easy scalability, accommodating increased loads or additional services with minimal reconfiguration.
- Security: JWT authentication and environment-based configuration management enhance the security of user sessions and sensitive data.
- Geospatial Precision: The use of PostGIS and Django's GIS tools enables accurate and reliable geospatial data handling, critical for effective water management.
- Ease of Deployment: Dockerization simplifies the deployment process, ensuring consistency across development, testing, and production environments.

Technologies Used:

- **Backend Framework:** Django 4.0
- **API Framework:** Django REST Framework
- **Authentication:** JWT via `rest_framework_simplejwt`
- **Geospatial Data Management:** PostgreSQL with PostGIS, `django.contrib.gis`
- **Containerization:** Docker, Docker-Compose
- **Email Services:** SMTP via Django
- **PDF Generation:** ReportLab