

Kyle Rego

Current role: Independent Software Developer

regoky@outlook.com • +1 (508) 813-8530 • github.com/KyleRego • kylerego.net

Work Experience

Loader / Unloader - Hub - United Parcel Service
Nov 2024 – Current

- Maintained punctuality and readiness for work each day
- Executed strenuous physical tasks safely
- Ensured safe handling of packages throughout the system

Delivery Driver – Domino's Pizza
Apr 2025 – Current

- Prepared food orders in a fast-paced kitchen
- Maintained cleanliness by washing dishes, sanitizing surfaces, sweeping, and mopping
- Handled cash transactions and ensured vehicle readiness for deliveries

Software Engineer (Web) - Impero Software
June 2022 – Feb 2024

- Collaborated across two scrum teams doing Ruby on Rails and ASP.NET Core development, in addition to working in a development support team that addressed production issues in a school focused SaaS product
- Designed and implemented a district dashboard feature for seamless user permission and device access schedule management across all schools in the district, optimizing efficiency for district administrators
- Engineered a support tool to merge duplicate student data securely, enhancing user experience by eliminating duplicates

Education

Rensselaer Polytechnic Institute *Bachelor of Science (4.0) Physics*
2018

Snippets

Anki Books - ankibooks.io

Ruby on Rails, Bootstrap, PostgreSQL, SQLite, Linux

- Developed Anki Record, a Ruby gem that provides an API for creating Anki deck packages (.apkg files), which are ZIP archives containing SQLite databases formatted to Anki's schema
- Engineered a Rails 7 backend to support Anki deck management, integrating with Anki Record to create downloadable .apkg files from the user's Anki Books data
- Designed a comprehensive test suite using RSpec for unit tests and Cucumber/Selenium/Capybara for end-to-end UI testing, ensuring robust and maintainable code

Larder - larder.lol

React, Bootstrap, ASP.NET Core, EF Core, SQLite, Linux

- Explored and implemented Controller-Service-Repository architecture in an ASP.NET Core backend and a React/TypeScript frontend to enforce separation of concerns
- Developed unit tests for services and repositories and Selenium UI tests to ensure maintainability and reliability