🛡️ Project Case Study: Odyssey Insurance Solution

(Laravel + React | Financial ERP System for Insurance)

📘 Project Overview

Built a **comprehensive Insurance ERP system** tailored for the U.S. insurance industry, empowering **investment management, account operations, check lifecycle automation**, and seamless communication with external carriers.
This robust, enterprise-grade platform was designed to automate and digitize the full policy lifecycle — from **investment handling and financial transaction logging to renewal reminders and secure document handling via FSSI**.

The solution replaced fragmented manual workflows with a unified, scalable, and secure platform — reducing operational overhead and drastically improving financial data accuracy.

🎯 Business Goals

- Automate end-to-end **check lifecycle management** including printing, voiding, reissue, and stale checks
- Streamline **investment and account center** operations with accurate financial entries
- Enable **real-time synchronization with carriers** (Farmers & Foremost) via agency/agent feeds
- Generate **ACH & Positive Pay files** with secure transmission
- Design a **renewal notification system** with email automation using AWS SES
- Enable scalable infrastructure with **containerization and background job handling**
- Provide secure **agent onboarding** via portal with approval workflows
- Maintain **dynamic branded templates** for companies under different profiles

---

🧑‍💻 My Role:

Senior Full Stack Engineer & System Architect

- Led the entire development lifecycle from architecture, backend, frontend, and DevOps
- Owned third-party integrations and deployment strategy

- Defined scalable logic for modules like check printing, ACH, and GL postings

---

🧰 Tech Stack

Backend: Laravel 10+, MySQL, Queues, Jobs, Task Scheduling, Eloquent ORM
Frontend: React.js, Tailwind CSS, Axios
DevOps & Infra: Docker, AWS EC2, AWS SES, Laravel Forge, GitHub Actions
Monitoring: CloudWatch, DataDog
External Services: FSSI (Document Processing), Bank Systems (Positive Pay, ACH)

⚙️ Core Modules & Key Functionalities

💼 Investment & Account Management

- **Investment Center**: Built from scratch to manage financial instruments, returns, and investment logs
- **Account Center**: Included support for accounts management, general entries, and memorized transactions for repeat accounting patterns
- **GL Posting**: Automated general ledger entries triggered by events (e.g., check issue, ACH transaction)

💸 Check Lifecycle Automation

- **Print Check Module**: Designed an automated check printing system with layout customization, print queues, and permission controls
- **Check Register System**: Enabled voiding, reissuing, and tracking stale checks with complete lifecycle logs and reporting

🧾 ACH, Banking & Security

- **Positive Pay File Generation**: Developed logic to export check metadata into bank-compliant Positive Pay format
- **ACH File Processor**: Enabled generation and secure transfer of ACH files for automated payments and disbursements

📨 Communication & Reminder Engine

- **Renewal Reminder Notices**: Scheduled batch job to send reminders for upcoming policy renewals via AWS SES

- **Email Templating System**: Role-based templates with dynamic data for personalized reminders

🧾 Agent & Carrier Integrations

- **Agent Enrollment Workflow**: Built a secure onboarding system with email verification, document upload, and admin approval
- **Carrier Feeds**: Integrated external sync logic for Farmers and Foremost carrier data feeds, including agent, agency, and policy sync

🏢 Dynamic Branding

- **Branded Company Profile Setup**: Enabled organizations to configure theme, logos, templates, and layout preferences to reflect brand identity

---

🔒 DevOps, Security & Observability

- **Containerized with Docker**: Seamless development-to-production parity and scaling
- **AWS SES Integration**: High-deliverability email communications (renewals, agent approvals, status updates)
- **Cloud Monitoring**: Used **AWS CloudWatch** and **Datadog** to monitor logs, performance, and system alerts
- **Queues & Scheduling**: All batch operations (feeds, renewals, ACH) were processed via Laravel jobs and schedulers

---

📈 Results & Impact

✅ Reduced manual operations by over **75%** through automation of check and ACH processing

✅ Improved **data accuracy and compliance** across financial modules and GL reporting

✅ Cut **email delivery time by 50%** with SES + templated batch notifications

✅ Onboarded **multiple carriers and 1000+ agents** with zero data inconsistency

✅ Deployed scalable microservice architecture using Docker & Laravel queues

✅ Delivered the MVP in **under 60 days** with production-grade stability

---

💡 What I Learned

- Mastered **complex financial logic** implementation using Laravel's event-driven structure
- Gained in-depth experience with **ACH/Positive Pay banking standards** and integrations
- Worked hands-on with **external document processing (FSSI)** and secure file transmissions
- Achieved seamless scalability using **queues, jobs, and Dockerized deployments**
- Deepened DevOps experience with **GitHub Actions, AWS SES, and CloudWatch**

📌 Key Takeaways

- Laravel queues + jobs = perfect for scalable batch ops like ACH & renewal reminders
- Docker + Forge + CI/CD = painless deployment and faster time to market
- Strong separation of finance modules = better code maintainability & accuracy
- React + Tailwind = fast, responsive UI with minimal overhead

---

🔗 Access & Demo

- 🔐 Code Access: Private repo (available for review under NDA)
- 🎯 Live Demo: Secured demo access can be arranged during interview sessions

---

🔚 Conclusion

The **Odyssey Insurance Solution** exemplifies my ability to **architect and deliver complex, enterprise-level systems** with financial precision, high scalability, and tight third-party integrations.
It proves my strength in **Laravel + React full-stack development, DevOps automation, and financial domain workflows**.
If you're looking for someone to handle **ERP-grade projects, custom SaaS systems, or insurance tech with sensitive data and integrations**, I'd love to collaborate.

Let's build something impactful together!
