



# PROJECT PORTFOLIO

Matt Gallam

## Abstract

This is a brief description of few of the software projects I've been a part of. I have explained the quick overview of the project, Use Cases, Implementation Highlights, and some prominent solutions I implemented for the problems ahead.

## Table of Contents

1. Modern Documentation Automation System for Medical Devices (Name: Authorea).....	2
Introduction .....	2
Use Case .....	2
Tech Stack.....	2
Implementation Highlights .....	2
Challenges and Solutions .....	2
Conclusion .....	3
2. Modern Financial Services Management System (Name: My Secure Advantage).....	4
Introduction .....	4
Use Case .....	4
Tech Stack.....	4
Project Overview .....	4
Challenges and Solutions .....	5
Conclusion .....	6
3. Customized Reservation System for University Amenities (Name: GNAR) .....	7
Introduction .....	7
Use Case .....	7
Tech Stack.....	7
Project Overview .....	7
Challenges and Solutions .....	8
Conclusion .....	9

# 1. Modern Documentation Automation System for Medical Devices (Name: Authorea)

## Introduction

In the dynamic landscape of the medical industry, efficient documentation is critical, especially when experimenting with new medical devices. Our project aimed to revolutionize the documentation process by developing a modern web application that automates the creation, mapping, and contextual suggestions for medical device documentation.

## Use Case

The primary use case addressed was streamlining the documentation process for medical devices. The system facilitated the automatic generation of comprehensive documentation, allowing users to map information from various source documents to destination documents. Users could also receive context-aware suggestions for document content, enhancing the overall efficiency of the documentation workflow.

## Tech Stack

- **Backend:** Python Django with a microservices architecture.
- **Frontend:** React for a responsive and user-friendly interface.
- **Cloud Provider:** AWS, utilizing EC2 for hosting, RDS for database hosting, and various services including Route53 and VPC.
- **Database:** PostgreSQL for robust data management.
- **Containerization:** Docker for seamless deployment and scalability.
- **Test Automation:** Cypress for end-to-end testing, ensuring system reliability.
- **CI/CD:** GitHub Actions for continuous integration and continuous deployment on EC2 instances.
- **Natural Language Processing:** Fine-tuned BioBERT for context-aware suggestions.
- **ETL Pipeline:** Leveraged Apache Airflow for efficient Extract, Transform, Load operations.
- **Vector Database:** Pinecone for optimized vector storage.

## Implementation Highlights

- **User Authentication:** OAuth 2.0 was implemented for secure user authentication.
- **Document Handling:** Users could upload documents in various formats (pdf or docx).
- **Mapping and Context Suggestions:** TinyMCE facilitated mapping of documents, and BioBERT provided context-aware suggestions.
- **ETL Pipeline:** Apache Airflow handled the ETL pipeline, converting documents to a standardized template.
- **Image Handling:** For documents with low OCR accuracy, the system saved images with absolute coordinates for precise mapping.
- **Scalability:** Airflow and asynchronous modules ensured scalability for simultaneous document uploads, even with varying sizes and formats.

## Challenges and Solutions

- OCR Accuracy

**Challenge:** Achieving 100% accuracy with OCR was challenging.

**Workaround:** Saving images alongside HTML files for inaccurate OCR sections, allowing users to pinpoint and map accurately.

- LLM Training Document Consistency

**Challenge:** Inconsistency in training documents' templates.

**Workaround:** Implemented an ETL pipeline to standardize documents before training the LLM, ensuring a consistent template.

- Scalable Architecture

**Challenge:** Designing a scalable architecture to handle varying document loads.

**Solution:** Leveraged Apache Airflow and asynchronous modules for seamless scalability, accommodating different document upload scenarios.

- Security and Compliance

**Challenge:** Ensuring the security and compliance of sensitive medical information.

**Solution:** Implemented strict access controls, encryption protocols, and regular security audits. Adhered to industry-specific compliance standards, such as HIPAA, to guarantee data integrity and confidentiality.

- Real-time Collaboration

**Challenge:** Facilitating real-time collaboration among multiple users working on the same document.

**Solution:** Integrated collaborative features, allowing simultaneous editing and real-time updates. Implemented version control to track changes and maintain document integrity.

- Integration with Legacy Systems

**Challenge:** Coordinating with existing legacy systems within medical institutions.

**Solution:** Developed robust APIs for seamless integration with legacy systems. Ensured backward compatibility and data synchronization to enhance interoperability.

- Data Volume and Storage Management

**Challenge:** Handling a large volume of diverse documents and managing storage efficiently.

**Solution:** Implemented an optimized data storage strategy, utilizing AWS S3 for scalable object storage. Integrated data archiving and lifecycle policies to manage storage costs effectively.

- Cross-Browser Compatibility

**Challenge:** Ensuring consistent performance and user experience across various web browsers.

**Solution:** Conducted rigorous cross-browser testing during development. Utilized responsive design principles to ensure compatibility with popular browsers, enhancing the application's accessibility.

- Dynamic Document Templates

**Challenge:** Adapting to evolving medical documentation standards and templates.

**Solution:** Implemented a dynamic template system that could be updated and customized based on emerging standards. Ensured that the ETL pipeline could handle changes in document structures without disrupting the workflow.

- Load Balancing

**Challenge:** Balancing server loads during peak usage times.

**Solution:** Employed AWS Elastic Load Balancing to distribute incoming application traffic across multiple EC2 instances. Configured auto-scaling to dynamically adjust resources based on demand, ensuring optimal performance.

- User Feedback and Iterative Improvements

**Challenge:** Continuously improving the system based on user feedback and evolving industry requirements.

**Solution:** Established a feedback loop with users, regularly collecting input and suggestions. Implemented an agile development approach, allowing for iterative updates and enhancements to address evolving needs.

## Conclusion

The Modern Documentation Automation System for Medical Devices successfully addressed the complexities of medical device documentation. The combination of cutting-edge technologies and strategic workarounds resulted in a robust and scalable solution, enhancing efficiency in the medical industry's documentation processes.

## 2. Modern Financial Services Management System (Name: My Secure Advantage)

### Introduction

In response to the evolving needs of a services-based company, our team developed a modern web application to streamline service tracking, revenue generation analysis, and client targeting. The project aimed to replace cumbersome manual processes, involving Excel sheets and separate ledgers, with an automated and sophisticated system.

### Use Case

The primary use case involved the efficient tracking of services provided to clients, revenue generation analysis, and targeted client acquisition based on industries contributing more revenue. The system also integrated seamlessly with QuickBooks for ledger maintenance and employed predictive analysis models to boost sales.

### Tech Stack

- **Backend:** Python Flask for a lightweight and scalable backend.
- **Frontend:** React for a responsive and intuitive user interface.
- **Database:** PostgreSQL for efficient data management.
- **Containerization:** Docker for streamlined deployment and scalability.
- **Test Automation:** Cypress for end-to-end testing, ensuring system reliability.
- **CI/CD:** GitHub Actions for continuous integration and continuous deployment on EC2 instances.
- **Monitoring:** Sentry for real-time error tracking.
- **ETL Pipeline:** Apache Airflow for a robust and scalable Extract, Transform, Load process.
- **Cloud Provider:** AWS, leveraging services like AWS Glue and AWS Athena for data processing and querying.
- **Predictive Analysis:** Utilized pandas, scikit-learn, and numpy with a Random Forest Regressor model for sales predictions.

### Project Overview

- ETL Pipeline:

Developed a robust ETL pipeline using Apache Airflow to aggregate data from Excel sheets.

Utilized AWS Glue for unstructured data storage in a Data Lake, and AWS Athena for efficient querying.

Data was seeded into PostgreSQL by calculating commissions, payments, ledger maintenance via QuickBooks, sub-contracting information, and geographical locations of service provision.

- Microservices Architecture:

Implemented a microservices architecture to ensure user authentication, hierarchical access, and data analytics.

Established a structured hierarchy to control user access based on roles, enabling a secure and controlled environment.

- Analytics and Predictive Analysis:

Integrated analytics modules to provide insights into services, team performance, industry preferences, and individual commissions.

Employed predictive analysis models using scikit-learn to boost sales by predicting future trends based on historical data.

- Seamless Integration with QuickBooks:

Established a seamless integration with QuickBooks for ledger maintenance and accurate financial record-keeping.

Ensured consistency and reliability in financial data by synchronizing with QuickBooks in real-time.

## Challenges and Solutions

- Data Accuracy and Consistency:

**Challenge:** Ensuring zero errors, especially in financial calculations.

**Solution:** Implemented rigorous data validation checks within the ETL pipeline to maintain accuracy. Conducted regular reconciliation checks to ensure consistency between different data sources.

- Real-time Integration with QuickBooks:

**Challenge:** Achieving real-time synchronization with QuickBooks for ledger maintenance.

**Solution:** Leveraged QuickBooks APIs and designed an efficient middleware for real-time data exchange, ensuring up-to-date and accurate financial records.

- User Adoption and Training:

**Challenge:** Overcoming user resistance to new systems and ensuring smooth adoption.

**Solution:** Conducted comprehensive training sessions, provided user-friendly documentation, and incorporated intuitive UI/UX elements to enhance user experience and minimize the learning curve.

- Scalability and Performance:

**Challenge:** Ensuring the system's scalability and optimal performance, especially during peak usage times.

**Solution:** Implemented containerization with Docker for scalability and utilized AWS services for load balancing. Regularly monitored system performance and optimized database queries for enhanced responsiveness.

- Model Training and Adaptability:

**Challenge:** Training and adapting predictive analysis models for changing market trends.

**Solution:** Implemented a continuous learning approach, regularly updating and retraining models based on new data. Established feedback loops to capture market changes and improve model accuracy over time.

- Data Security and Compliance:

**Challenge:** Ensuring the security and compliance of sensitive financial data.

**Solution:** Implemented robust access controls, encryption protocols, and conducted regular security audits. Adhered to industry-specific compliance standards to guarantee data integrity and confidentiality.

- Client Data Privacy:

**Challenge:** Safeguarding client data privacy and adhering to strict data protection regulations.

**Solution:** Implemented encryption protocols for data at rest and in transit. Conducted regular privacy assessments and ensured compliance with data protection laws, such as GDPR, to maintain client trust and legal adherence.

- Third-Party Integration Complexity:

**Challenge:** Managing the complexity of integrating with third-party services, especially QuickBooks.

**Solution:** Conducted in-depth analysis and testing of QuickBooks APIs. Developed a modular and flexible integration architecture to accommodate changes in third-party systems. Established error handling mechanisms for seamless data exchange.

- User Interface Complexity:

**Challenge:** Balancing feature-rich interfaces with user-friendly design.

**Solution:** Conducted user experience (UX) studies to understand user preferences. Implemented a clean and intuitive design with progressive disclosure, allowing users to access advanced features progressively. Regularly gathered user feedback for iterative UI improvements.

- Handling Historical Data:

**Challenge:** Managing and integrating historical data from Excel sheets into the new system.

**Solution:** Developed data migration scripts to handle the transition from legacy Excel sheets to the new system. Ensured that historical data was accurately mapped and maintained during the ETL process.

- Data Duplication and Inconsistency:

**Challenge:** Preventing data duplication and ensuring consistency across different databases and systems.

**Solution:** Implemented unique identifiers and data validation checks during the ETL process to avoid duplication. Employed referential integrity constraints within the database schema to maintain data consistency.

- Scalable Predictive Analysis:

**Challenge:** Scaling predictive analysis models as the business expands and data volume increases.

**Solution:** Employed distributed computing frameworks, such as Apache Spark, for scalable model training. Implemented model versioning and automated retraining processes to ensure the models stayed relevant and accurate with the growth of the business.

- User Access Control Complexity:

**Challenge:** Managing complex user access control hierarchies while maintaining system performance.

**Solution:** Implemented a role-based access control (RBAC) system with fine-grained permissions. Conducted regular performance optimizations to ensure that user access controls did not compromise system responsiveness.

- Adapting to Regulatory Changes:

**Challenge:** Adapting the system to evolving financial regulations and industry standards.

**Solution:** Established a dedicated regulatory compliance team to monitor changes in financial regulations. Maintained a flexible system architecture that allowed for quick updates to comply with new regulatory requirements.

## Conclusion

The Modern Financial Services Management System successfully addressed the complex challenges of the financial industry by providing an integrated, automated, and predictive solution. The combination of advanced technologies, seamless integrations, and strategic solutions ensures that the system meets the dynamic needs of the services company, ultimately leading to enhanced operational efficiency and strategic decision-making.

# 3. Customized Reservation System for University Amenities (Name: GNAR)

## Introduction

The NRS Reservation System aimed to streamline the reservation process for University amenities such as rooms, auditoriums, grounds, and libraries. The project involved the development of a customized calendar system, manager and staff portals, authorization mechanisms, and a calculation module to optimize the reservation workflow. Additionally, the system included a data migration process to ensure the seamless transition from the legacy system to the new platform.

## Use Case

The primary use case involved creating a tailored reservation system for University amenities. Each amenity had its own customized calendar, and the system facilitated reservation calculations, manager oversight, and a user-friendly experience for staff and administrators.

## Tech Stack

- **Backend:** Python Flask for a lightweight and scalable backend.
- **ORM:** SQL Alchemy for efficient database operations.
- **Frontend:** React for a responsive and intuitive user interface.
- **Containerization:** Docker for streamlined deployment and scalability.
- **Cloud Hosting:** AWS for reliable and scalable hosting.

## Project Overview

- **Customized Reservation Calendar:**

Implemented a personalized calendar system for each reserved amenity.

Each amenity had its own dedicated calendar, allowing users to view, reserve, and manage bookings efficiently.

- **Manager and Staff Portals:**

Developed dedicated portals for both managers and general staff members.

Managers had access to comprehensive oversight tools, while staff members could easily navigate and manage reservations through a user-friendly interface.

- **Authorization Mechanisms:**

Implemented robust authorization mechanisms for managers and administrators.

Ensured that access levels were well-defined, maintaining data security and integrity throughout the system.

- **Calculation Module for Reserves:**

Integrated a calculation module to optimize the reservation process.

Automated the calculation of reservation durations, costs, and other relevant parameters, streamlining the administrative workload.

- **Data Migration:**

Successfully migrated and validated data from the legacy reservation system to the new platform.

Ensured data consistency and accuracy during the migration process, minimizing disruptions in reservation services.

- **Dockerization and AWS Hosting:**

Containerized the application using Docker for consistent deployment across different environments.

Hosted the system on AWS for scalability, reliability, and efficient management of resources.



## Challenges and Solutions

- Customization Complexity:

**Challenge:** Implementing unique calendar systems for each reserved amenity.

**Solution:** Employed a modular design approach, allowing for dynamic creation of calendars based on amenity types. Implemented a flexible configuration system to handle diverse reservation requirements.

- Authorization Security:

**Challenge:** Designing and implementing a secure authorization system.

**Solution:** Utilized industry-standard authentication and authorization practices. Conducted regular security audits to identify and address potential vulnerabilities.

- Calculation Accuracy:

**Challenge:** Ensuring accurate calculations for reservation durations and costs.

**Solution:** Implemented rigorous testing of the calculation module, including edge cases. Conducted regular audits to verify the accuracy of reservation-related calculations.

- Data Migration Complexity:

**Challenge:** Migrating and validating data from the legacy system without disruptions.

**Solution:** Developed robust data migration scripts, conducted extensive testing, and performed a phased migration to minimize downtime and ensure data integrity.

- Scalability for Increased Reservations:

**Challenge:** Scaling the system to accommodate increased reservation demands.

**Solution:** Designed the system with scalability in mind, utilizing AWS resources effectively. Employed load testing to identify potential bottlenecks and optimized performance accordingly.

- Scalability for Future Growth:

**Challenge:** Anticipating and preparing for the system's scalability as the university infrastructure and reservation demands grow.

**Solution:** Implemented horizontal scaling strategies, ensuring that the application could efficiently handle an increasing number of concurrent reservations. Utilized AWS auto-scaling features to dynamically adjust resources based on demand, ensuring optimal performance during peak usage periods.

- Availability and Redundancy:

**Challenge:** Guaranteeing high availability and minimizing downtime for critical reservation services.

**Solution:** Utilized AWS services like Elastic Load Balancing (ELB) for distribution of incoming traffic across multiple instances, enhancing system availability. Implemented redundancy measures, such as multi-AZ deployments, to ensure continuous service availability even in the event of a server failure.

- Clean Code and Software Development Life Cycle (SDLC):

**Challenge:** Enforcing clean code practices and a structured SDLC for long-term maintainability.

**Solution:** Instituted code quality standards and conducted regular code reviews to ensure adherence to clean code principles. Established a well-defined SDLC with distinct phases for planning, coding, testing, and deployment. Integrated automated code analysis tools to identify and address code smells and maintain consistency in coding styles.

- Database Optimization for Efficiency:

**Challenge:** Maintaining efficient database performance as the volume of reservation data increases.

**Solution:** Regularly optimized database queries and indexes to ensure quick and responsive data retrieval. Utilized database partitioning strategies to manage and organize large datasets effectively, enhancing overall system efficiency.

- Proactive Monitoring and Issue Resolution:

**Challenge:** Identifying and addressing potential issues proactively before they impact system performance.

**Solution:** Implemented robust monitoring tools, including AWS CloudWatch and custom logging solutions. Established automated alerting systems to notify administrators of potential issues. Conducted regular performance audits and preemptive maintenance to address concerns before they escalated.

- Documentation for Long-Term Understanding:

**Challenge:** Ensuring comprehensive and up-to-date documentation for long-term understanding and maintenance.

**Solution:** Maintained detailed documentation for codebase architecture, APIs, and deployment procedures. Conducted regular documentation reviews to reflect changes in system functionality. Encouraged developers to document code changes thoroughly to facilitate knowledge transfer and onboarding of new team members.

## Conclusion

The Customized Reservation System for University Amenities successfully addressed the intricacies of managing diverse amenities, providing an efficient and user-friendly solution for reservation processes. The strategic implementation of technology and systematic handling of challenges ensured a seamless transition from the legacy system to a modern and scalable platform.