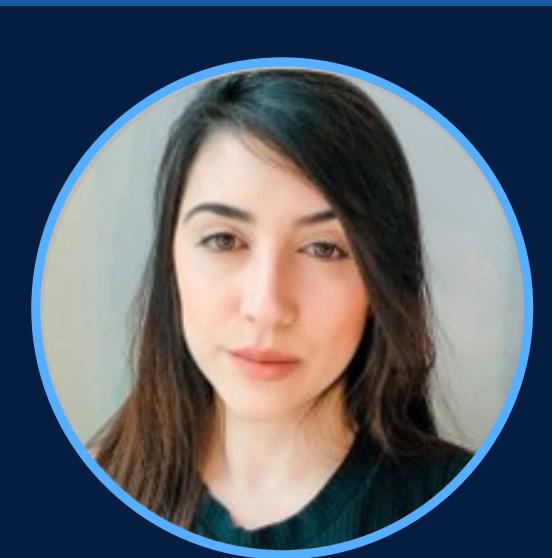
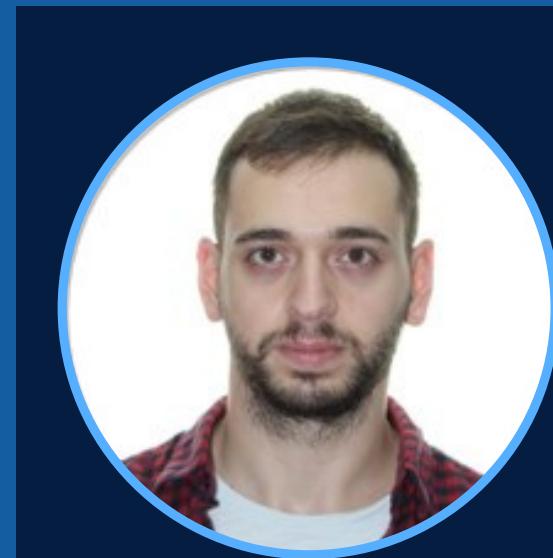




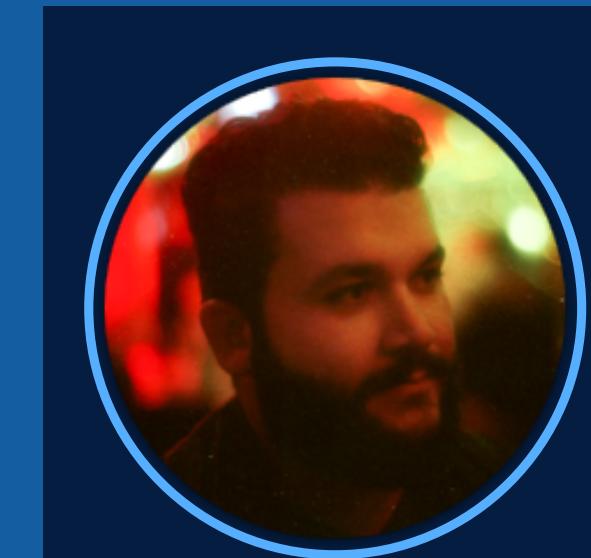
OUR TEAM



Jaque
Duarte
101400994



Gorkem
Sari
101266503



Igor
Oliveira
101375347



Valunchanut
Simaroj
101386344

PROJECT DESCRIPTION

"The AI-IIIMS project aims to revolutionize inventory management in the retail sector. Designed with a focus on efficiency, cost-effectiveness, and empowering decision-making, AI-IIIMS offers retail clients a sophisticated tool for optimizing restocking processes. This initiative promises not only to streamline operations but also to ensure significant cost savings, enabling businesses to navigate the complexities of inventory management with confidence and precision."



PROJECT COMPONENTS

Data Cleaning & Acquisition

- Gather raw data from multiple XLSX files, ensuring a comprehensive collection of daily sales information.
- Read and parse XLSX files to organize data into a consistent format with days as columns, product names as rows, and sales figures as cell values.

Tools: Python, Pandas, OpenPyXL

Forecasting

- Utilize the Prophet library to forecast daily sales or ingredient usage based on cleaned data.
- Adjust data structures to fit the forecasting model's requirements, ensuring accurate temporal predictions.

Tools: Python, Numpy, Pandas, Prophet

API Layer

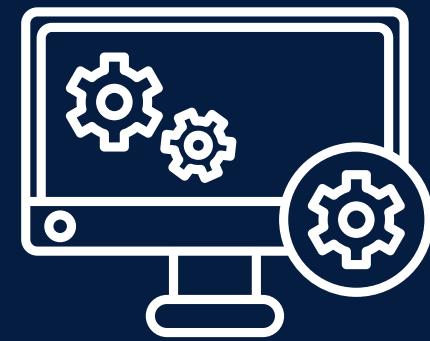
- Develop endpoints to retrieve specific product sales or ingredient usage and to compute forecasts over specified date ranges.
- Prepare and format data for front-end consumption, ensuring seamless integration with the web interface.

Tools: Python, FastAPI

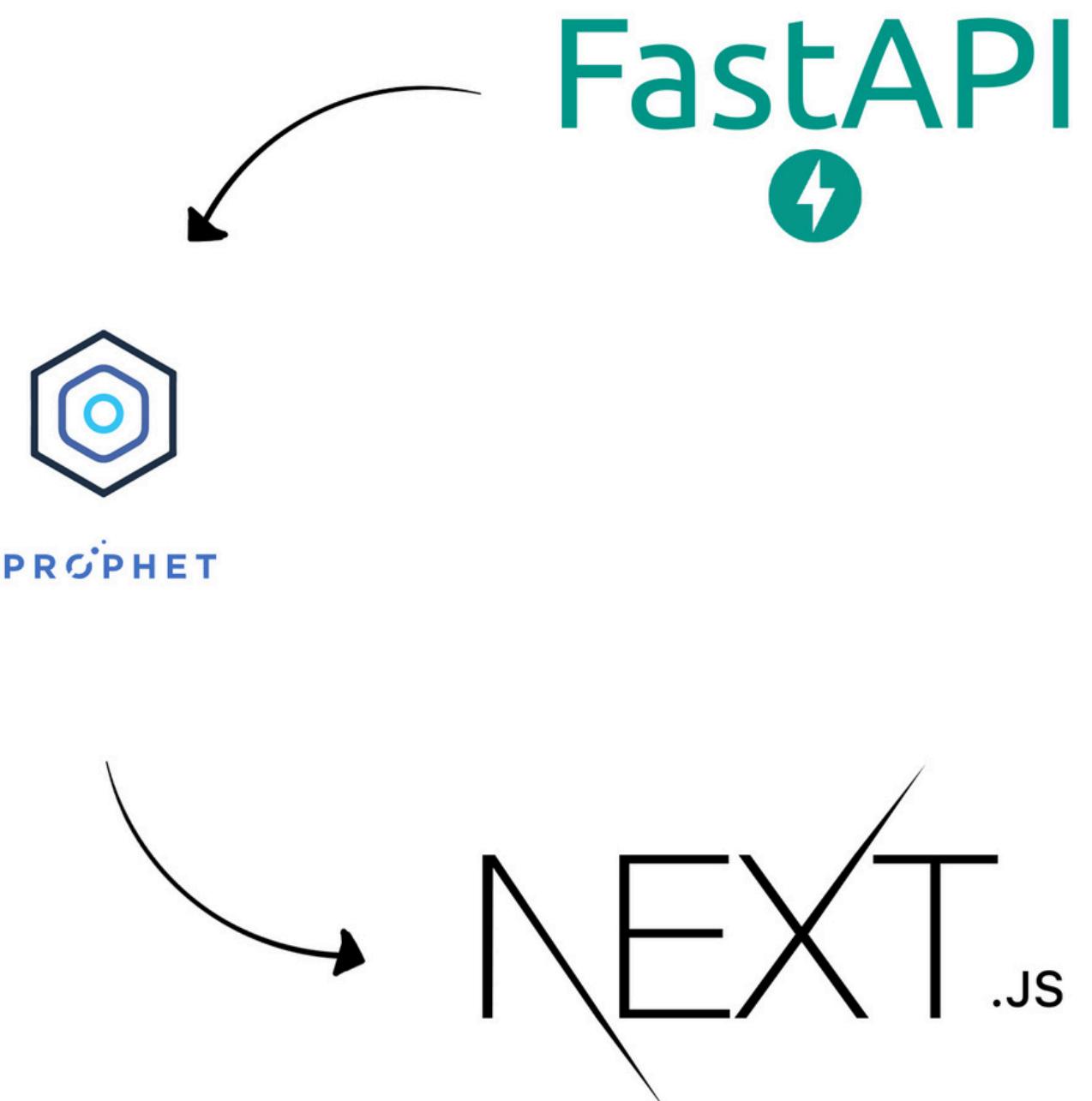
Frontend

- Build a responsive web interface using to allow users to select products or ingredients and specify date ranges for data visualization.
- Display data through dynamic charts and graphs.

Tools: Next.js, React.js, Tailwind CSS



TECHNOLOGIES USED



Data Cleaning

- **Pandas (Python Library):**

- **Functionality:** Pandas offers robust data structures and operations designed to manipulate numerical tables and time series efficiently.
- **Application:** Used in the Inventory Management System to clean, transform, and aggregate complex sales data from multiple Excel files. It handles tasks such as sorting dates, aggregating sales figures, and transforming datasets into a more analyzable format.
- **Benefits:** Ensures accurate and efficient data handling, reduces errors, and streamlines the data cleaning process. It supports handling large datasets with ease, providing a solid foundation for subsequent forecasting tasks.

- **Openpyxl (Python Library):**

- **Functionality:** Enables reading and writing Excel files without needing Microsoft Excel installed, focusing on working with .xlsx files.
- **Application:** Utilized to read sales data directly from Excel files in the Inventory Management System, allowing for the extraction of necessary data fields directly from the source.
- **Benefits:** Facilitates direct interaction with Excel files, enhancing the flexibility to access and manipulate spreadsheet data. It is especially useful for projects where data is initially collected in Excel formats, ensuring seamless data import and preprocessing.

Forecasting

- **Pandas (Python Library):**
 - **Functionality:** As described previously, Pandas facilitates data manipulation and analysis. In forecasting, it helps in filtering data by dates and preparing it for the forecasting model.
 - **Application:** Used for pre-processing data before feeding it into the Prophet model, ensuring the data fits the model's requirements, such as resetting indices and renaming columns for compatibility.
 - **Benefits:** Pandas streamlines the preparation of data for forecasting, enhancing the accuracy and efficiency of the process.
- **NumPy (Python Library):**
 - **Functionality:** Provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
 - **Application:** Integrated within the forecasting function to handle operations like conditional adjustments to forecasted values.
 - **Benefits:** Ensures robust numerical computation within the forecasting workflow, enhancing data integrity and the usability of forecast results.

Forecasting

- **Prophet (Python Library):**

- **Functionality:** Developed by Facebook, Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
- **Application:** In the Inventory Management System, Prophet is utilized to forecast future sales based on historical sales data. The tool models trends in sales data to predict future sales figures for various products within specified date ranges.
- **Benefits:** Prophet simplifies the forecasting process, especially with time series that have strong seasonal effects and several seasons of historical data. It handles missing data and shifts in trends well, making it robust against common problems in real-world data.

API Layer

- **FastAPI (Python Framework):**
 - **Functionality:** FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.6+ based on standard Python type hints.
 - **Application:** Utilized to create RESTful interfaces for the Inventory Management System, FastAPI handles HTTP requests for forecasting product sales. It provides a robust solution to accept requests, process them, and return forecast data.
 - **Benefits:** FastAPI supports asynchronous request handling, which increases the performance of the application. It automatically generates interactive API documentation (using Swagger UI) that helps in testing and validating API endpoints efficiently.

Frontend

- **Next.js (React Framework):**

- **Functionality:** Next.js is a powerful React framework that enables functionality such as server-side rendering and generating static websites for React-based web applications.
- **Application:** The framework handles the web application's structure, providing pages and routing mechanisms that facilitate the display and interaction with the forecast data provided by the backend API.
- **Benefits:** Next.js enhances the user experience by improving the performance through server-side rendering, which reduces the load times and initial page speeds. Additionally, it supports automatic code splitting, ensuring that only necessary code loads for each page, further enhancing performance.

- **React.js (JavaScript Library):**

- **Functionality:** React.js is a library for building user interfaces, particularly known for its virtual DOM that optimizes the rendering process.
- **Application:** Integrated within Next.js, React.js allows developers to construct reusable UI components that manage their state and render efficiently as data changes, crucial for displaying the forecast data dynamically.
- **Benefits:** React's component-based architecture makes the frontend scalable and maintainable. It helps in building large web applications where data can change over time without reloading the page, ensuring a seamless user experience.

Frontend

- **Styling and Layout:**

- **CSS and Frameworks:** Tailored CSS alongside frameworks like Tailwind CSS or Bootstrap could be utilized to style the application. These tools provide utility classes and responsive design features to quickly design modern and mobile-friendly interfaces.
- **Application:** Styling is crucial for making the interface user-friendly and visually appealing. For instance, responsive tables to display sales forecasts, interactive charts for trend analysis, and intuitive forms for date range selection.
- **Benefits:** Using CSS frameworks accelerates the development process, promotes design consistency, and ensures that the application is accessible on various devices and screen sizes.

Version Control

- **Git/Github:**

- **Functionality:** Git is a distributed version control system that allows multiple developers to work on a project simultaneously without interfering with each other's work. GitHub is a cloud-based hosting service that leverages Git's version control capabilities and adds its own features.
- **Application:** For the Inventory Management System, GitHub serves as the central repository that hosts the source code, tracks changes, manages pull requests, and facilitates issue tracking and collaboration among developers.
- **Benefits:** The use of GitHub enhances team collaboration through features such as branch management, pull requests for code review, and integration with tools like GitHub Actions for continuous integration and deployment. This ensures a streamlined workflow and efficient management of project updates.



SCOPES



IN SCOPE

- 01** Inventory: The system will display current inventory, so the manager can calculate the difference to plan future orders
- 02** Sales Forecasting: The system will forecast future sales based on past sales
- 03** Detailed system design, which will be covered in subsequent project documentation.
- 04** Cross-platform web application accessible via standard web browsers.

OUT OF SCOPE

- 01** Procurement and deployment of hardware infrastructure.
- 02** Point of Sale (POS) System: IMS will not include POS functionality.
- 03** Accounting: Accounting and financial modules are not part of IMS
- 04** Integration of the AI-IIMS with existing inventory management systems and databases.
- 05** Data upload functionalities for sales and inventory
- 06** Inventory Tracking: IMS will track inventory levels, sales, deliveries, and waste.
- 07** Detailed system design, which will be covered in subsequent project documentation.
- 08** Payment module
- 09** Admin module for modifying database related and other critical components.



LESSONS LEARNED

- To have access for the selling point before proposal (data).
- Create visual documents explaining the process of the app.
- Better communication and asking instead of assuming.



BEST PRACTICES

- Maintaining comprehensive documentation throughout the project lifecycle.
- Utilizing pair programming to enhance code quality and knowledge transfer.
- Breaking down the project work into manageable tasks, allowing team members to focus and complete their sections individually.
- Learning proper data formats.
- Using Prophet as a provider for forecasting.
- Being able to use what we learned in with SCRUM/AGILE



Thank you :)



PROJECT TIMELINE

