**DEVELOPMENT: FULL NAMES AND COMPONENT REGISTRY**

1. **System name:** Optimized Multi-Restaurant Ordering System
2. **Components:**

* **Frontend Module (React & JavaScript):** The user interface for the web platform is developed using React and JavaScript, providing a seamless and interactive experience for consumers to place orders across multiple restaurants.
* **Backend Module (Python & Flask):** The server-side logic, including order management and processing, is handled by Python with Flask as the web framework. This ensures efficient communication between the frontend and the database, managing requests from customers and relaying them to the respective restaurants.
* **Database (MongoDB):** A centralized NoSQL database (MongoDB) is used to store information related to restaurants, menus, orders, tables, and customer billing details. MongoDB allows for flexibility in managing large datasets across different restaurants in real-time.
* **API Integration (Flask & JavaScript**): Flask is used to create APIs that allow the frontend (React) to communicate with the backend and the database, enabling seamless data flow between the client interface and restaurant systems.
* **Order Notification and Management System (Python & Flask):** Orders are processed and managed through Python's backend, with notifications sent to the relevant restaurants along with details of the table and the customer’s bill for preparation and delivery.

**ABOUT THE PROJECT**

The system is designed to enhance the experience of consumers in food courts. It allows users to place orders from different restaurants through a single web interface without needing to physically visit each establishment. Each restaurant receives the order with the customer's bill and table details, enabling them to prepare the order and deliver it directly to the specified table. The primary goal is to optimize the ordering process and improve customer convenience.