إدارة وتخطيط المشروع ((Project Planning & Management

1.1 هيكل فريق العمل ((Team Structure & Roles

	المهام الأساسية	الدور
Mostafa Ashraf	إدارة المشروع، إعداد بيئة Docker، وإدارة الـ Kafka Broker.	**Team Leader & DevOps**
Basmala AbuElhamd	تدريب موديلات LSTM و chronos، وتطوير نظام YOLOv8 للفحص البصري.	**AI & Vision**
Ibrahim Yussif	تصميم وإدارة قواعد البيانات (PostgreSQL & InfluxDB) ونظام الأرشفة (Parquet).	**Database Architect**
Mohammed Hamed	تطوير محاكي الحساسات (Sensor Simulator) وإدارة تدفق البيانات عبر كافكا.	**Streaming Engineer**
Mahmmud Mohammed	بناء الـ FastAPI، وتطوير منطق الـ What-If Analysis للتحليل التنبؤي.	**Backend Developer**
Ola Alaa	تصميم لوحات تحكم Grafana وربط التنبيهات اللحظية واجهة المستخدم.	**Visualization**

1.2 الجدول الزمني للمشروع ((Project Timeline

- **المرحلة الأولى:** إعداد البنية التحتية وجمع البيانات التاريخية
- **المرحلة الثانية:** تدريب الموديلات الذكية (ML & Vision) وإعداد كافكا
- **المرحلة الثالثة:** تطوير طبقة الـ API وربط قواعد البيانات
- **المرحلة الرابعة:** تصميم الـ Dashboards والاختبار النهائي

Literature Review

يمثل مشروع Digital Twin for CNC Machin تطبيقاً عملياً لمفاهيم الثورة الصناعية الرابعة (Industry 4.0)، حيث تتكامل تقنيات مراقبة الحالة (Condition Monitoring)، والتعلم الآلي (Machine Learning)، والرؤية الحاسوبية

(Computer Vision)، ومعالجة البيانات اللحظية (Stream Processing) لخلق نظام ذكي قادر على التنبؤ بالأعطال قبل وقوعها. في هذا القسم، سيتم استعراض الأسس النظرية والمقاربات السابقة التي تم البناء عليها في هذا المشروع.

2.1 التشخيص والتنبؤ في الأنظمة الصناعية ((Diagnostic and Prognostic in Industrial Systems

يلعب نشاط الصيانة دوراً رئيسياً في الأنظمة الصناعية حيث يسمح بتحسين التوافر، والموثوقية، والأمان، مع تقليل تكلفة دورة الحياة. يمكن تصنيف الصيانة إلى فئتين رئيسيتين: الصيانة العلاجية (Curative) والصيانة الوقائية (Preventive).

تعتبر الصيانة المبنية على الحالة (CBM - Condition Based Maintenance) من أهم أنواع الصيانة الوقائية. في هذا النهج، يتم مراقبة حالة النظام الصناعي بشكل مستمر وفحصها بواسطة مجموعة من المستشعرات (Sensors). يتم بعد ذلك معالجة البيانات التي تسجلها هذه المستشعرات لاستخراج الميزات (Features) ذات الصلة التي تسمح بتقدير الحالة الصحية الحالية وتوقع تطورها في المستقبل.

1+

- **التشخيص (Diagnostic):** يهدف إلى تقييم الحالة الحالية للمكون وتحديد سبب الفشل أو العطل.
- **التنبؤ (Prognostic):** يستخدم للتنبؤ بالحالة الصحية المستقبلية للمكون لتوقع وتجنب الفشل قبل حدوثه.

بشكل رسمي، يتكون التنبؤ بالفشل من تقدير الوقت المتبقي قبل الفشل، أو ما يُعرف بالعمر الإنتاجي المتبقي (Remaining RUL - Useful Life)، وقيمة الثقة المرتبطة به (Confidence Value). ويعرف معيار ISO 13381-1 التنبؤ بأنه تقدير وقت التشغيل قبل الفشل وخطر وجود أو ظهور وضع أو أكثر من أوضاع الفشل في المستقبل.

2.2 Data-Driven Prognostic Approach

يمكن تحقيق التنبؤ من خلال ثلاث طرق رئيسية: التنبؤ القائم على النموذج (Model-based)، التنبؤ القائم على الخبرة (Experience-based)، والتنبؤ القائم على البيانات (Data-driven).

يعتمد مشروعنا بشكل أساسي على **النهج القائم على البيانات (Data-driven prognostic)**. يهدف هذا النهج إلى تحويل بيانات المراقبة الخام (Raw monitoring data) التي توفرها المستشعرات إلى نماذج سلوكية ذات صلة للنظام، بما في ذلك التدهور والاهتراء (Degradation).

يُفضل هذا النهج في الحالات التي يكون فيها الحصول على بيانات موثوقة من المستشعرات أسهل من بناء نماذج سلوكية تحليلية معقدة (كما في التنبؤ القائم على النماذج) أو الانتظار للحصول على بيانات خبرة كافية. في الممارسة العملية، معظم ظواهر التدهور (مثل تآكل أداة القطع في ماكينات CNC) تكون معقدة بسبب عدم الخطية (Non-linearities) والطبيعة العشوائية (Stochasticity) وصعوبة نمذجتها بدقة باستخدام النماذج التحليلية البحتة.

تُستخدم عدة أدوات ضمن النهج القائم على البيانات، معظمها مشتق من الذكاء الاصطناعي، ومنها:

- الشبكات العصبية ((Neural Networks
- (Kalman Filter)
- نماذج ماركوف (Hidden Markov Models - HMM)

- **نماذج التعلم الآلي:** مثل XGBoost و LSTM

2.3 تقدير العمر الإنتاجي المتبقي (RUL) - Remaining Useful Life)

يعد حساب الـ RUL هو الهدف النهائي لأي نظام صيانة تنبؤية. ترتبط قيمة الـ RUL المُقدرة بشكل وثيق بالتقييم المستمر لحالة النظام (في حالتنا، تآكل الأداة في ماكينة الـ CNC). كما ذُكر في الورقة البحثية الخاصة بتشخيص وتوقع تآكل أدوات ماكينات CNC، فإن الخطأ المرتبط بتقدير الـ RUL يجب أن يتناقص كلما اقترب وقت الفشل الحقيقي، وذلك لأن التنبؤات يتم تعديلها في كل مرة يتم فيها اكتساب بيانات جديدة.

بالإضافة إلى ذلك، يجب أن ترتبط درجة ثقة (Confidence Degree) بقيمة الـ RUL المُقدرة. على سبيل المثال، من الأكثر واقعية وفعالية أن نبلغ مدير المصنع أن الماكينة ستفشل خلال فترة زمنية معينة مع مستوى ثقة معين (مثلاً 95%) بدلاً من إعطاء رقم دقيق قد لا يكون مضموناً.

2.4 قواعد البيانات والبيانات التاريخية المستخدمة ((Datasets Utilized

لضمان دقة النماذج، تم الاعتماد على قواعد بيانات قياسية وموثوقة لتدريب النماذج ومحاكاة السيناريوهات:

1. **CNC Mill Tool Wear Dataset:** تم استخدام هذه البيانات المستقاة من تجارب عملية (مثل تحدي PHM Society Data Challenge) لنمذجة سلوك تآكل أداة القطع. توفر هذه البيانات قراءات مستمرة من مستشعرات مختلفة (مثل القوة، الاهتزاز، الانبعاث الصوتي) أثناء عملية القطع. في دراسات مشابهة، تم استخدام هذه البيانات لتصنيف مراحل التآكل (Wear Stages) وتدريب نماذج التنبؤ بالـ RUL، وهي تتوافق مع محاكي الحساسات (Sensor Simulator) الذي يقوم بإرسال بيانات 50 حساساً.
https://github.com/makinarocks/awesome-industrial-machine-datasets/tree/master/data-explanation/CNC%20Mill%20Tool%20Wear#data-download-link
2. **MVTec AD Dataset:** تم استخدام قاعدة البيانات هذه لتدريب واختبار موديل الرؤية الحاسوبية (YOLOv8). تُعد MVTec AD معياراً مرجعياً قوياً (Benchmark) لاكتشاف العيوب (Anomaly Detection) في البيئات الصناعية، حيث تحتوي على صور عالية الدقة لمنتجات متعددة مع تصنيفات دقيقة للعيوب السطحية، مما يتيح للنظام إجراء فحص بصري موثوق (Visual Inspection) لحظياً.
https://www.kaggle.com/datasets/ipythonx/mvtec-ad

3. **Productivity Prediction of Garment Employees Dataset:** على الرغم من أن تركيزنا الأساسي هو ماكينات CNC، إلا أن تضمين بيانات متعلقة بالإنتاجية يوفر بعداً إضافياً لنظام تحليل "ماذا لو" (What-If Analysis). يسمح ذلك بربط حالة الماكينة بالإنتاجية العامة لفريق العمل، مما يوفر رؤية شاملة لمدير المصنع على لوحة تحكم Grafana.
https://archive.ics.uci.edu/dataset/597/productivity+prediction+of+garment+employees

2.5 استراتيجية دمج المراقبة اللحظية والرؤية الحاسوبية

في هذا المشروع، نتجاوز الاعتماد التقليدي على مستشعرات الماكينة فقط، بدمج **الرؤية الحاسوبية (Computer Vision)** كطبقة إضافية للتحقق من جودة المنتج النهائي. يتم استخدام خوارزميات مثل **YOLOv8** لمعالجة إطارات الصور لحظياً (Real-time Vision Process) لاكتشاف العيوب المرئية.

يتم دمج هذا التدفق المرئي مع تدفق بيانات المستشعرات اللحظية عبر **Apache Kafka**. هذا الدمج المعماري (كما يظهر في الـ DFD الخاص بالمشروع) يسمح بإنشاء قاعدة بيانات وصفية (Meta Data) في **PostgreSQL** تربط بين تنبيهات صحة الماكينة اللحظية (Machine Health Alerts) وبين نتيجة الفحص البصري للمنتج في نفس اللحظة الزمنية. هذا الترابط يوفر لمدير المصنع فهماً عميقاً وشاملاً لحالة النظام، مدعوماً بصور فعلية للعيوب معروضة على واجهة الاستخدام (Grafana Dashboard).

Requirements Gathering

3.1 Stakeholder Analysis

1. **مدير المصنع (Factory Manager):**
   a. **الاحتياجات:** رؤية شاملة وعالية المستوى لحالة الماكينات، تقليل تكاليف التوقف المفاجئ، وزيادة كفاءة الإنتاج.
   b. **الاستفادة من النظام:** استخدام لوحة تحكم (Grafana) لمراقبة مؤشرات الأداء، واستخدام ميزة "تحليل ماذا لو" (What-If Analysis) لاختبار سيناريوهات تشغيل مختلفة وتأثيرها على عمر الماكينة.
2. **مهندس الصيانة (Maintenance Engineer):**
   a. **الاحتياجات:** بيانات دقيقة وتنبؤات مبكرة بأعطال أداة القطع (Tool Wear) لجدولة الصيانة الوقائية.
   b. **الاستفادة من النظام:** استقبال تنبيهات لحظية بناءً على مخرجات نماذج الذكاء الاصطناعي (LSTM & XGBoost) توضح العمر الإنتاجي المتبقي (RUL).
3. **مشغل الماكينة (Machine Operator):**
   a. **الاحتياجات:** التحقق من جودة المنتج أثناء عملية التشغيل دون إيقاف الماكينة.
   b. **الاستفادة من النظام:** التنبيه الفوري في حال اكتشاف نظام الرؤية الحاسوبية (YOLOv8) لأي عيوب بصرية (Visual Defects) في المنتج.

4. **مهندس النظم/البيانات (Data Engineer): / System Admin)**
   a. **الاحتياجات:** نظام مستقر، قابل للتوسع، وسهل المراقبة.
   b. **الاستفادة من النظام:** بنية تحتية تعتمد على Docker و Kafka تضمن تدفق البيانات بسلاسة وتخزينها بشكل منظم (PostgreSQL & InfluxDB).

3.2 Use Cases

لتوضيح كيفية تفاعل المستخدمين مع النظام في سيناريوهات واقعية، تم صياغة قصص المستخدمين التالية:

- **User Story1 (المراقبة اللحظية):** "بصفتي مهندس صيانة، أريد أن أرى قراءات الحساسات الـ 50 (السرعة، الاهتزاز، الطاقة) تتحدث لحظياً على الشاشة، لكي أتمكن من ملاحظة أي شذوذ في أداء الماكينة فور حدوثه."
- **User Story 2 (التنبؤ بالأعطال):** "بصفتي مدير مصنع، أريد أن يعرض لي النظام تنبيهاً يوضح أن أداة القطع ستتلف بعد 15 ساعة عمل، لكي أقوم بطلب قطع غيار جديدة وتجنب توقف خط الإنتاج."
- **User Story 3 (الفحص البصري):** "بصفتي مشغل الماكينة، أريد أن يقوم النظام بفحص الكاميرا تلقائياً وتلوين إطار الصورة بالأحمر إذا كان هناك عيب سطحي في المنتج، لكي أقوم باستبعاده فوراً."
- **User Story 4 (تحليل What-if):** "بصفتي مخطط إنتاج، أريد إدخال سرعة قطع (Feedrate) جديدة في النظام الافتراضي، لكي أتوقع كيف ستؤثر هذه السرعة على العمر الافتراضي للماكينة قبل تطبيقها في الواقع."

3.3 المتطلبات الوظيفية (Functional Requirements)

وهي قائمة بالوظائف والميزات التقنية الأساسية التي يجب أن يؤديها النظام:

1. **استقبال البيانات اللحظية (Data Ingestion):** يجب أن يكون النظام قادراً على استقبال بيانات من محاكي المستشعرات (Edge Simulator) بمعدل فكل 50كل 50-100 مللي ثانية عبر Apache Kafka.
2. **معالجة الصور (Image Processing):** يجب أن يقوم النظام باستقبال إطارات الصور من الكاميرا وتمريرها لموديل YOLO لاكتشاف العيوب لحظياً.
3. **التحليل التنبؤي (Predictive Analysis):** يجب أن تقوم نماذج (LSTM و Transformer) بتحليل السلاسل الزمنية للحساسات وتحديث قيمة العمر المتبقي (RUL) بصفة دورية.
4. **واجهة برمجة التطبيقات (API Layer):** يجب توفير واجهة خلفية باستخدام FastAPI تستقبل طلبات الـ (What-If Analysis) وتعيد التوقعات بناءً على معطيات المستخدم.
5. **إدارة التخزين (Data Storage):**
   a. تخزين البيانات السريعة (Time-series) في InfluxDB.
   b. تخزين البيانات الوصفية (Metadata) ونتائج الفحص البصري في PostgreSQL.
   c. أرشفة البيانات التاريخية في ملفات Parquet لتسهيل إعادة تدريب الموديلات.
6. **العرض والتنبيهات (Visualization & Alerts):** يجب أن يقوم النظام بعرض البيانات على لوحات تحكم Grafana وإصدار تنبيهات بصرية للمستخدمين.

3.4 المتطلبات غير الوظيفية (Non-functional Requirements)

وهي المعايير التي تضمن جودة وأداء النظام بشكل احترافي:

1. **الأداء والسرعة (Performance):**
   a. يجب ألا يتجاوز زمن الاستجابة (Latency) لمعالجة بيانات الحساسات وعرضها على لوحة التحكم 100 مللي ثانية.

b. يجب أن تتم معالجة الفحص البصري (Inference Time) لـ YOLOv في زمن يسمح بمواكبة سرعة الإنتاج.

2. **الموثوقية والتوافر (Reliability & Availability):**

يجب أن تعمل البنية التحتية (Docker Containers لـ Kafka وقواعد البيانات) بنسبة توافر لا تقل عن 99% (Uptime).

a. يجب توفير آلية للتعافي (Fault Tolerance) بحيث لا تضيع البيانات في حالة توقف مؤقت لأحد الـ Consumers.

3. **قابلية التوسع (Scalability):**

a. يجب أن يدعم التصميم المعماري إضافة ماكينات CNC جديدة (New Producers/Topics) مستقبلاً دون الحاجة لإعادة كتابة الكود الأساسي للمشروع.

4. **الأمان (Security):**

a. حماية واجهات الـ API بحيث لا يُسمح بإرسال أوامر "ماذا لو" إلا للمستخدمين المصرح لهم.

b. تأمين قواعد البيانات بكلمات مرور وصلاحيات دخول محددة (Role-based Access).

5. **سهولة الاستخدام (Usability):**

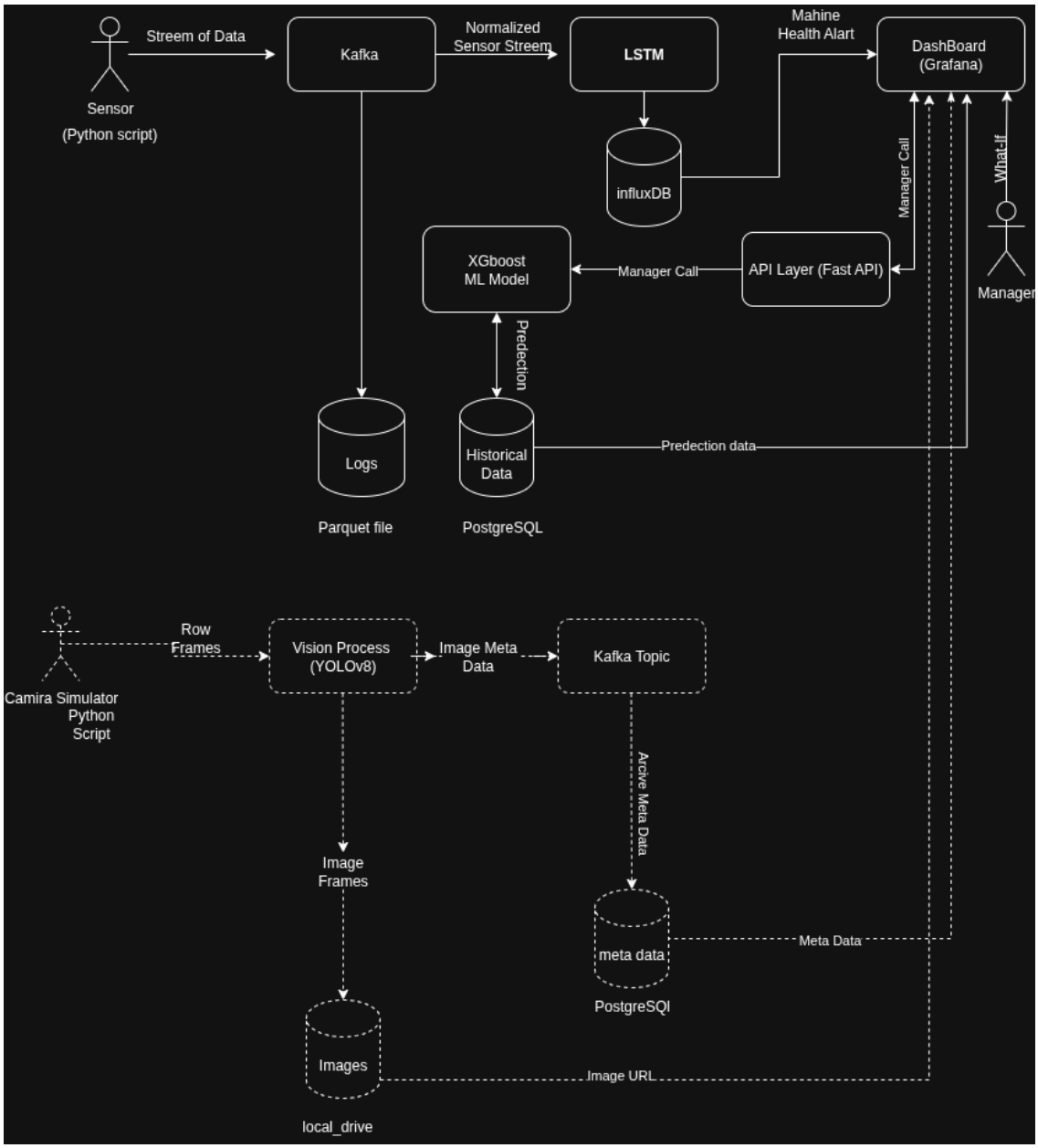
a. يجب أن تكون لوحات تحكم Grafana واضحة، تعتمد على الألوان (أخضر للحالة السليمة، أحمر للخطر)، ومفهومة للمستخدم غير التقني (مثل مدير المصنع).

System Analysis & Design

4.1 Problem Statement & Objectives

- **Problem Statement:** تعتمد المصانع التقليدية على أساليب "الصيانة التفاعلية" (Reactive Maintenance) أو "الصيانة المجدولة" (Scheduled Maintenance)، مما يؤدي إلى توقف مفاجئ وغير مخطط للماكينات (Unplanned Downtime) بسبب تآكل أداة القطع، بالإضافة إلى ارتفاع نسبة العيوب التصنيعية نتيجة غياب الفحص اللحظي أثناء التشغيل.
- **Objectives:** بناء "توأم رقمي" (Digital Twin) يعتمد على بيانات المستشعرات اللحظية لتمكين "الصيانة التنبؤية" (Predictive Maintenance)، وتقييم جودة المنتج باستخدام الرؤية الحاسوبية (Computer Vision)، وتوفير بيئة افتراضية لاختبار سيناريوهات التشغيل (What-If Analysis) لزيادة كفاءة الماكينة وتقليل التكاليف.

Software Architecture 4.2



DFD For System

يعتمد النظام على **Event-Driven Microservices Architecture**، وهي البنية الأنسب للتعامل مع Streaming Data. يتكون النظام من طبقات رئيسية:

1. **Edge Layer:** محاكيات الماكينة (Sensor Simulator & Camera Simulator) التي تولد البيانات.
2. **Message Broker Layer:** استخدام Apache Kafka لاستقبال وبث البيانات بسرعات عالية (High Throughput).
3. **Processing & AI Layer:** مKafka Consumers يقوم بتنظيف البيانات وتمريرها لنماذج (LSTM, Transformers, YOLOv8) للحصول على التنبؤات اللحظية.

4. **API Layer:** مبنية باستخدام FastAPI لربط واجهات العرض بمحركات التحليل وتوفير خدمة (What-If).
5. **Presentation Layer :** واجهات المستخدم المبنية على Grafana.

4.3 Database Design & Data Modeling

نظراً لطبيعة البيانات المزدوجة في المشروع، تم اعتماد تصميم Hybrid Data Architecture:

- **المخطط المنطقي والمادي (Logical & Physical Schema):**
  - **قاعدة بيانات السلاسل الزمنية (InfluxDB):** مصممة خصيصاً لاستقبال قراءات المستشعرات (السرعة، الاهتزاز، الطاقة) حيث يكون "الزمن" (Timestamp) هو المفتاح الأساسي.
  - **قاعدة البيانات العلائقية (PostgreSQL):** تستخدم لإدارة الكيانات الثابتة والبيانات الوصفية.
- **مخطط العلاقات (Entity-Relationship Diagram - ER Diagram):** يحتوي الـ ERD الخاص بـ PostgreSQL على الجداول التالية:
  - جدول Machines: (رقم الماكينة، النوع، الحالة).
  - جدول Operators: (رقم المشغل، الاسم، الصلاحية).
  - جدول Vision_Inspections: (رقم الفحص، رابط الصورة المرجعي Image_URL، حالة الفحص: ناجح/معيب، تاريخ الفحص).
  - جدول Maintenance_Logs: (سجلات الصيانة، حالة الأداة، العمر المتبقي المقدر RUL).

4.4 تصميم واجهة المستخدم والنمذجة (UI/UX Design & Prototyping)

تعتمد واجهة المستخدم بشكل أساسي على **Grafana**، وتم وضع المبادئ التوجيهية التالية (UI/UX Guidelines) لتصميم لوحات التحكم:

- **Wireframes & Mockups:** تنقسم الشاشة الرئيسية إلى ثلاثة أجزاء:
  - Scorecardsتوضح الحالة العامة (صحي، تحذير، خطر) والعمر المتبقي (RUL) بنسبة مئوية.
  - *الجزء الأوسط:* رسوم بيانية خطية (Line Charts) حية توضح تذبذب قراءات الحساسات مع خطوط تحذيرية حمراء (Thresholds).
  - *الجزء الجانبي:* عرض لآخر صورة التقطتها الكاميرا مع نتيجة الفحص البصري (Passed/Failed).
- **المبادئ البصرية:** استخدام نظام ألوان مريح للعين وسريع التنبيه (الأخضر للعمليات الطبيعية، الأصفر للتنبيهات، الأحمر للحالات الحرجة التي تتطلب تدخلاً فورياً).

4.6 System Deployment & Integration

- **Technology Stack:**
  - *Backend :* Python, FastAPI.
  - *Broker:* Apache Kafka.
  - *Databases:* PostgreSQL, InfluxDB, Parquet Files.
  - *AI:* LSTM (TensorFlow/Keras), chronos-forecasting, Ultralytics (YOLOv8).

- *Frontend:* Grafana.
- **Deployment & Component :**

يتم نشر جميع مكونات النظام باستخدام **Docker Compose**. حيث يمثل كل مكون (Postgres, Kafka, Zookeeper, FastAPI, Grafana) حاوية مستقلة (Container) تعمل داخل شبكة افتراضية معزولة (Docker Bridge Network) لضمان سهولة الإعداد، التشغيل، والتكامل السلس بين