```
-- النتائج العامة
overall_grade ENUM('Excellent', 'Good', 'Fair', 'Poor', 'Rejected'),
quality_score DECIMAL(5,2), -- من 0 إلى 100
compliance_standards JSON, -- معايير الامتثال المحققة
defects_found JSON, -- العيوب المكتشفة
improvement_recommendations TEXT,

-- شهادات الجودة
certification_eligible BOOLEAN,
export_approved BOOLEAN,
premium_grade BOOLEAN,
organic_certified BOOLEAN,

-- التتبع والمرجعية
batch_number VARCHAR(50),
traceability_code VARCHAR(100),
chain_of_custody JSON, -- سلسلة الحضانة

-- التكاليف والسعر
analysis_cost DECIMAL(8,2),
market_price_adjustment DECIMAL(5,2), -- تعديل السعر بناءً على الجودة

-- المتابعة
retest_required BOOLEAN DEFAULT FALSE,
retest_date DATE,
corrective_actions JSON,
follow_up_notes TEXT,

approved_by VARCHAR(100),
approval_date DATE,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

);

17. **جدول الذكاء الاصطناعي والتحليلات (AI_Analytics_Results)**

```sql
CREATE TABLE ai_analytics_results (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    analysis_type ENUM('production_prediction', 'health_risk_assessment', 'breeding_optimization',
                       'feed_efficiency', 'market_forecast', 'weather_impact', 'behavior_analysis'),
    entity_type ENUM('individual_animal', 'herd', 'farm', 'product_batch'),
    entity_id BIGINT,
    model_version VARCHAR(20),
    algorithm_used VARCHAR(100),

    -- بيانات الدخل
    input_data JSON, -- البيانات المستخدمة في التحليل
    data_quality_score DECIMAL(4,2), -- جودة البيانات المدخلة
    confidence_level DECIMAL(5,2), -- مستوى الثقة في النتائج

    -- النتائج والتوقعات
    predictions JSON, -- التوقعات المختلفة
    risk_factors JSON, -- عوامل المخاطر المحددة
    recommendations JSON, -- التوصيات المقترحة
    optimization_suggestions JSON, -- اقتراحات التحسين

    -- المقاييس الإحصائية
    accuracy_percentage DECIMAL(5,2),
    precision_score DECIMAL(5,2),
    recall_score DECIMAL(5,2),
    f1_score DECIMAL(5,2),

    -- الفترة الزمنية المتوقعة
    prediction_start_date DATE,
    prediction_end_date DATE,
    forecast_horizon_days INT,

    -- النتائج المالية المتوقعة
    expected_revenue_change DECIMAL(10,2),
    expected_cost_savings DECIMAL(10,2),
    roi_prediction DECIMAL(7,4),

    -- حالة التنفيذ
    implementation_status ENUM('pending', 'in_progress', 'completed', 'rejected'),
    actual_vs_predicted JSON, -- مقارنة النتائج الفعلية مع المتوقعة
```

```
    -- البيانات التقنية
    processing_time_seconds DECIMAL(8,3),
    computational_resources JSON,
    model_parameters JSON,

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

18. جدول إنترنت الأشياء والأجهزة الذكية (IoT_Devices_Management)

sql

```sql
CREATE TABLE iot_devices_management (
    id INT PRIMARY KEY AUTO_INCREMENT,
    device_id VARCHAR(50) UNIQUE NOT NULL,
    device_name VARCHAR(100),
    device_type ENUM('rfid_reader', 'gps_tracker', 'temperature_sensor', 'humidity_sensor',
                     'weight_scale', 'milking_sensor', 'feed_dispenser', 'water_monitor',
                     'camera', 'weather_station', 'gate_controller', 'alarm_system'),
    manufacturer VARCHAR(100),
    model VARCHAR(100),
    firmware_version VARCHAR(50),

    -- الموقع والتثبيت
    installation_location VARCHAR(200),
    gps_coordinates POINT,
    installation_date DATE,
    last_maintenance_date DATE,
    next_maintenance_date DATE,

    -- الاتصال والشبكة
    connection_type ENUM('wifi', 'ethernet', '4G', '5G', 'lora', 'zigbee', 'bluetooth'),
    network_address VARCHAR(50), -- IP أو MAC address
    communication_protocol VARCHAR(50),
    data_transmission_frequency INT, -- بالثواني

    -- الحالة التشغيلية
    current_status ENUM('online', 'offline', 'maintenance', 'error', 'low_battery'),
    last_seen DATETIME,
    uptime_percentage DECIMAL(5,2),
    error_count_24h INT,

    -- الطاقة والبطارية
    power_source ENUM('mains', 'battery', 'solar', 'hybrid'),
    battery_level INT, -- نسبة مئوية
    power_consumption_watts DECIMAL(6,2),
    battery_life_estimate_hours INT,

    -- البيانات والإعدادات
    sensor_configuration JSON, -- إعدادات المستشعرات
    alert_thresholds JSON, -- عتبات التنبيهات
    calibration_data JSON, -- بيانات المعايرة
    last_calibration_date DATE,

    -- الأمان والحماية
```

```
    encryption_enabled BOOLEAN DEFAULT TRUE,
    access_credentials JSON, -- بيانات الوصول المشفرة
    security_level ENUM('low', 'medium', 'high', 'critical'),
    last_security_update DATE,

    -- الصيانة والتشغيل
    warranty_expiry_date DATE,
    service_contract_number VARCHAR(50),
    replacement_cost DECIMAL(8,2),
    operational_cost_monthly DECIMAL(6,2),

    -- الأداء والإحصائيات
    data_accuracy_percentage DECIMAL(5,2),
    false_positive_rate DECIMAL(5,4),
    response_time_ms INT,
    throughput_data_per_hour DECIMAL(10,2),

    assigned_to_entity_type ENUM('livestock', 'facility', 'vehicle', 'general'),
    assigned_to_entity_id BIGINT,

    created_by INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

🎨 المميزات المتقدمة الإضافية (تفاصيل شاملة)

🧠 نظام الذكاء الاصطناعي المتطور

1. تحليلات التنبؤ الذكية (Predictive Analytics)

javascript

```
const predictiveAnalytics = {
 // التنبؤ بالإنتاج
 async predictMilkProduction(animalId, timeframe) {
  const historicalData = await this.getHistoricalProduction(animalId, 365);
  const weatherData = await this.getWeatherForecast(timeframe);
  const feedingData = await this.getCurrentFeedingPlan(animalId);
  const healthData = await this.getHealthStatus(animalId);

  const aiModel = new ProductionPredictionModel({
   historicalData,
   weatherData,
   feedingData,
   healthData,
   seasonalFactors: this.getSeasonalFactors(),
   breedCharacteristics: await this.getBreedData(animalId)
  });

  return {
   predictedDailyProduction: aiModel.predictDaily(),
   confidenceLevel: aiModel.getConfidenceScore(),
   factorsInfluence: aiModel.getInfluencingFactors(),
   recommendations: aiModel.getOptimizationSuggestions(),
   riskFactors: aiModel.identifyRiskFactors(),
   expectedRange: {
    minimum: aiModel.predictMinimum(),
    maximum: aiModel.predictMaximum(),
    mostLikely: aiModel.predictMostLikely()
   },
   timeSeriesData: aiModel.generateTimeSeriesForecast(timeframe)
  };
 },

 // تقييم المخاطر الصحية
 async healthRiskAssessment(animalId) {
  const vitalSigns = await this.getRealtimeVitalSigns(animalId);
  const behaviorPatterns = await this.getBehaviorAnalysis(animalId, 30);
  const environmentalData = await this.getEnvironmentalConditions();
  const geneticPredisposition = await this.getGeneticRiskFactors(animalId);

  const healthAI = new HealthRiskModel();

  return {
   overallRiskScore: healthAI.calculateOverallRisk(),
```

```
      specificRisks: {
        respiratoryInfection: healthAI.assessRespiratoryRisk(),
        metabolicDisorders: healthAI.assessMetabolicRisk(),
        reproductiveIssues: healthAI.assessReproductiveRisk(),
        nutritionalDeficiency: healthAI.assessNutritionalRisk(),
        behavioralStress: healthAI.assessStressRisk()
      },
      earlyWarningSignals: healthAI.identifyEarlyWarnings(),
      preventiveMeasures: healthAI.suggestPreventiveMeasures(),
      urgencyLevel: healthAI.determineUrgency(),
      recommendedActions: healthAI.generateActionPlan(),
      monitoringFrequency: healthAI.suggestMonitoringSchedule()
    };
  },

  // تحسين التكاثر الوراثي
  async breedingOptimization(femaleId, availableMales) {
    const femaleGenetics = await this.getGeneticProfile(femaleId);
    const maleGenetics = await Promise.all(
      availableMales.map(id => this.getGeneticProfile(id))
    );

    const breedingAI = new GeneticOptimizationModel({
      female: femaleGenetics,
      males: maleGenetics,
      breedingGoals: await this.getBreedingObjectives(),
      marketDemand: await this.getMarketTrends()
    });

    return {
      optimalMatches: breedingAI.rankBreedingPairs(),
      expectedOffspring: breedingAI.predictOffspringTraits(),
      geneticDiversity: breedingAI.assessGeneticDiversity(),
      economicValue: breedingAI.calculateEconomicValue(),
      recommendedTiming: breedingAI.suggestOptimalTiming(),
      successProbability: breedingAI.predictSuccessRate(),
      longTermImpact: breedingAI.assessLongTermGenetics()
    };
  }
};
```

تكامل Blockchain لسلسلة التوريد

2. نظام التتبع اللامركزي

javascript

```
const blockchainTraceability = {
 // تسجيل معاملة جديدة على البلوك تشين
 async recordTransaction(transactionData) {
  const blockchainTx = {
   timestamp: new Date().toISOString(),
   transactionId: this.generateUniqueId(),
   entityType: transactionData.type, // livestock, product, etc.
   entityId: transactionData.entityId,
   action: transactionData.action, // birth, transfer, processing, sale
   location: transactionData.location,
   parties: {
    from: transactionData.fromParty,
    to: transactionData.toParty,
    intermediaries: transactionData.intermediaries || []
   },
   certifications: transactionData.certifications || [],
   qualityData: transactionData.qualityData || {},
   environmentalData: transactionData.environmentalData || {},
   previousHash: await this.getPreviousBlockHash(transactionData.entityId),
   digitalSignature: await this.createDigitalSignature(transactionData),
   verificationData: {
    rfidScans: transactionData.rfidScans || [],
    gpsCoordinates: transactionData.gpsCoordinates,
    photographicEvidence: transactionData.images || [],
    witnessSignatures: transactionData.witnesses || []
   }
  };

  // تسجيل على البلوك تشين
  const blockHash = await this.submitToBlockchain(blockchainTx);

  // حفظ محلياً للمرجعية السريعة
  await this.saveLocalReference({
   blockHash,
   transactionId: blockchainTx.transactionId,
   entityId: transactionData.entityId,
   timestamp: blockchainTx.timestamp
  });

  return {
   success: true,
   blockHash,
   transactionId: blockchainTx.transactionId,
```

```
      verificationUrl: this.generateVerificationUrl(blockHash)
    };
  },

  // التحقق من صحة سلسلة التوريد
  async verifySupplyChain(productId, fromOrigin = true) {
    const chainData = await this.getCompleteChain(productId, fromOrigin);

    const verification = {
      isValid: true,
      totalSteps: chainData.length,
      verificationResults: [],
      timeline: [],
      qualityHistory: [],
      locationHistory: [],
      ownershipHistory: [],
      certificationsHistory: []
    };

    for (let step of chainData) {
      const stepVerification = await this.verifyBlockchainTransaction(step.blockHash);
      verification.verificationResults.push(stepVerification);

      if (!stepVerification.isValid) {
        verification.isValid = false;
      }

      verification.timeline.push({
        timestamp: step.timestamp,
        action: step.action,
        location: step.location,
        verified: stepVerification.isValid
      });
    }

    return verification;
  }
};
```

🌡️ نظام البيئة الذكية والاستدامة

3. مراقبة الأثر البيئي

javascript

```
const environmentalMonitoring = {
 // حساب البصمة الكربونية
 async calculateCarbonFootprint(farmId, timeframe) {
  const livestockData = await this.getLivestockData(farmId, timeframe);
  const feedData = await this.getFeedConsumption(farmId, timeframe);
  const energyData = await this.getEnergyConsumption(farmId, timeframe);
  const transportData = await this.getTransportData(farmId, timeframe);
  const wasteData = await this.getWasteProduction(farmId, timeframe);

  const carbonCalculator = new CarbonFootprintCalculator();

  return {
   totalCO2Equivalent: carbonCalculator.calculateTotal(),
   breakdown: {
    livestockEmissions: carbonCalculator.calculateLivestockEmissions(livestockData),
    feedProduction: carbonCalculator.calculateFeedEmissions(feedData),
    energyConsumption: carbonCalculator.calculateEnergyEmissions(energyData),
    transportation: carbonCalculator.calculateTransportEmissions(transportData),
    wasteManagement: carbonCalculator.calculateWasteEmissions(wasteData),
    landUseChange: carbonCalculator.calculateLandUseEmissions()
   },
   comparisonToBenchmark: carbonCalculator.compareToIndustryBenchmark(),
   reductionOpportunities: carbonCalculator.identifyReductionOpportunities(),
   offsetRecommendations: carbonCalculator.suggestOffsetStrategies(),
   sustainabilityScore: carbonCalculator.calculateSustainabilityScore(),
   certificationEligibility: carbonCalculator.assessCertificationEligibility()
  };
 },

 // مراقبة جودة المياه
 async monitorWaterQuality(farmId) {
  const waterSources = await this.getWaterSources(farmId);
  const qualityResults = [];

  for (let source of waterSources) {
   const iotSensors = await this.getWaterQualitySensors(source.id);
   const latestReadings = await this.getLatestSensorReadings(iotSensors);

   const qualityAnalysis = {
    sourceId: source.id,
    sourceName: source.name,
    sourceType: source.type, // well, river, municipal, etc.
    timestamp: new Date(),
```

```
      parameters: {
        ph: latestReadings.ph,
        dissolvedOxygen: latestReadings.dissolvedOxygen,
        turbidity: latestReadings.turbidity,
        temperature: latestReadings.temperature,
        conductivity: latestReadings.conductivity,
        totalDissolvedSolids: latestReadings.tds,
        nitrates: latestReadings.nitrates,
        phosphates: latestReadings.phosphates,
        bacterialCount: latestReadings.bacterialCount,
        heavyMetals: latestReadings.heavyMetals || {}
      },
      qualityRating: this.assessWaterQuality(latestReadings),
      safetyStatus: this.assessWaterSafety(latestReadings),
      recommendations: this.generateWaterQualityRecommendations(latestReadings),
      treatmentNeeded: this.assessTreatmentNeeds(latestReadings),
      complianceStatus: this.checkRegulatoryCompliance(latestReadings)
    };

    qualityResults.push(qualityAnalysis);
  }

  return {
    overallWaterQuality: this.calculateOverallWaterQuality(qualityResults),
    individualSources: qualityResults,
    alerts: this.generateWaterQualityAlerts(qualityResults),
    trends: await this.analyzeWaterQualityTrends(farmId, 90),
    improvementPlan: this.generateWaterQualityImprovementPlan(qualityResults)
  };
 }
};
```

📱 تطبيق الواقع المعزز والمختلط

4. واجهة الواقع المعزز للعمليات الحقلية

javascript

```
const augmentedRealityInterface = {
 // عرض معلومات الحيوان عبر الواقع المعزز
 async displayAnimalInfo(cameraStream, animalDetection) {
  const animalId = await this.identifyAnimalFromImage(animalDetection);
  const animalData = await this.getComprehensiveAnimalData(animalId);

  const arOverlay = {
   animalIdentification: {
    rfidTag: animalData.rfidTag,
    name: animalData.name || `Animal ${animalData.id}`,
    breed: animalData.breed,
    age: this.calculateAge(animalData.birthDate),
    confidence: animalDetection.confidence
   },
   healthIndicators: {
    overallHealth: animalData.healthStatus,
    lastVaccination: animalData.lastVaccination,
    nextCheckup: animalData.nextCheckup,
    currentTemperature: animalData.realtimeData?.temperature,
    heartRate: animalData.realtimeData?.heartRate,
    alerts: animalData.activeAlerts || []
   },
   productionData: {
    dailyMilkProduction: animalData.production?.dailyMilk,
    monthlyAverage: animalData.production?.monthlyAverage,
    productionTrend: animalData.production?.trend,
    qualityRating: animalData.production?.qualityRating
   },
   behaviorAnalysis: {
    currentActivity: animalData.behavior?.currentActivity,
    stressLevel: animalData.behavior?.stressLevel,
    socialInteractions: animalData.behavior?.socialRank,
    abnormalBehaviors: animalData.behavior?.abnormalPatterns || []
   },
   locationData: {
    currentLocation: animalData.location?.current,
    preferredAreas: animalData.location?.preferredAreas,
    restrictedZones: animalData.location?.restrictedZones,
    lastSeen: animalData.location?.lastUpdate
   },
   quickActions: [
    { action: 'health_check', label: 'فحص صحي', icon: '🩺' },
    { action: 'feed_schedule', label: 'جدولة تغذية', icon: '🌾' },
```

```
      { action: 'vaccination', label: 'تطعيم', icon: '💉' },
      { action: 'weight_measure', label: 'قياس وزن', icon: '⚖️' },
      { action: 'breeding_plan', label: 'خطة تكاثر', icon: '💞' },
      { action: 'transfer', label: 'نقل', icon: '🚚' }
    ],
    visualCues: {
      healthIndicator: this.getHealthColorCode(animalData.healthStatus),
      productionIndicator: this.getProductionColorCode(animalData.production),
      attentionLevel: this.calculateAttentionLevel(animalData),
      boundingBox: animalDetection.boundingBox,
      trackingPath: animalData.movement?.recentPath || []
    }
  };

  return {
    overlayData: arOverlay,
    renderInstructions: this.generateARRenderInstructions(arOverlay),
    trackingData: this.generateTrackingData(animalDetection),
    interactionCapabilities: this.getAvailableInteractions(animalData)
  };
},

// عرض حالة المرافق عبر الواقع المعزز
async displayFacilityStatus(cameraStream, facilityDetection) {
  const facilityId = await this.identifyFacilityFromImage(facilityDetection);
  const facilityData = await this.getFacilityStatusData(facilityId);

  const arOverlay = {
    facilityInfo: {
      name: facilityData.name,
      type: facilityData.type, // barn, milking_parlor, feed_storage, etc.
      capacity: facilityData.capacity,
      currentOccupancy: facilityData.currentOccupancy,
      utilizationRate: (facilityData.currentOccupancy / facilityData.capacity) * 100
    },
    environmentalConditions: {
      temperature: facilityData.sensors?.temperature,
      humidity: facilityData.sensors?.humidity,
      airQuality: facilityData.sensors?.airQuality,
      lightLevel: facilityData.sensors?.lightLevel,
      noiseLevel: facilityData.sensors?.noiseLevel,
      ventilation: facilityData.systems?.ventilation?.status
    },
    equipmentStatus: facilityData.equipment?.map(eq => ({
```

```
      id: eq.id,
      name: eq.name,
      status: eq.status,
      efficiency: eq.efficiency,
      maintenanceStatus: eq.maintenanceStatus,
      alerts: eq.alerts || []
    })),
    alerts: facilityData.alerts?.filter(alert => alert.severity === 'high'),
    maintenanceSchedule: facilityData.maintenance?.upcoming,
    safetyIndicators: {
      fireSuppressionStatus: facilityData.safety?.fireSuppression,
      emergencyExitsStatus: facilityData.safety?.emergencyExits,
      securitySystemStatus: facilityData.safety?.securitySystem,
      structuralIntegrity: facilityData.safety?.structuralCheck
    }
  };

  return arOverlay;
 }
};
```

🤝 نظام إدارة العلاقات مع الشركاء (PRM)

5. إدارة العلاقات مع المزارع الفرعية

```
javascript
```

```
const partnerRelationshipManagement = {
 // نظام تقييم الأداء الشامل
 async evaluatePartnerPerformance(partnerId, evaluationPeriod) {
  const partnerData = await this.getPartnerData(partnerId);
  const performanceMetrics = await this.getPerformanceMetrics(partnerId, evaluationPeriod);
  const complianceData = await this.getComplianceRecords(partnerId, evaluationPeriod);
  const financialData = await this.getFinancialPerformance(partnerId, evaluationPeriod);
  const qualityData = await this.getQualityMetrics(partnerId, evaluationPeriod);

  const evaluation = {
   partnerId,
   evaluationPeriod,
   overallScore: 0,
   categories: {
    operational: {
     score: 0,
     metrics: {
      livestockCare: this.evaluateLivestockCare(performanceMetrics.livestock),
      facilityMaintenance: this.evaluateFacilityMaintenance(performanceMetrics.facilities),
      staffCompetency: this.evaluateStaffCompetency(performanceMetrics.staff),
      technologyAdoption: this.evaluateTechnologyAdoption(performanceMetrics.technology),
      processCompliance: this.evaluateProcessCompliance(complianceData.processes)
     }
    },
    financial: {
     score: 0,
     metrics: {
      profitability: this.evaluateProfitability(financialData.profit),
      costEfficiency: this.evaluateCostEfficiency(financialData.costs),
      paymentReliability: this.evaluatePaymentReliability(financialData.payments),
      revenueGrowth: this.evaluateRevenueGrowth(financialData.revenue),
      investmentInImprovements: this.evaluateInvestments(financialData.investments)
     }
    },
    quality: {
     score: 0,
     metrics: {
      productQuality: this.evaluateProductQuality(qualityData.products),
      animalWelfare: this.evaluateAnimalWelfare(qualityData.welfare),
      foodSafety: this.evaluateFoodSafety(qualityData.safety),
      traceability: this.evaluateTraceability(qualityData.traceability),
      certificationCompliance: this.evaluateCertifications(qualityData.certifications)
     }
```

```
    },
    relationship: {
      score: 0,
      metrics: {
        communication: this.evaluateCommunication(performanceMetrics.communication),
        responsiveness: this.evaluateResponsiveness(performanceMetrics.responsiveness),
        collaboration: this.evaluateCollaboration(performanceMetrics.collaboration),
        transparency: this.evaluateTransparency(performanceMetrics.transparency),
        conflictResolution: this.evaluateConflictResolution(performanceMetrics.conflicts)
      }
    },
    sustainability: {
      score: 0,
      metrics: {
        environmentalImpact: this.evaluateEnvironmentalImpact(performanceMetrics.environment),
        resourceEfficiency: this.evaluateResourceEfficiency(performanceMetrics.resources),
        wasteManagement: this.evaluateWasteManagement(performanceMetrics.waste),
        energyUsage: this.evaluateEnergyUsage(performanceMetrics.energy),
        carbonFootprint: this.evaluateCarbonFootprint(performanceMetrics.carbon)
      }
    }
  },
  strengths: [],
  areasForImprovement: [],
  actionPlan: [],
  incentiveEligibility: {},
  contractRenewalRecommendation: '',
  riskAssessment: {
    overallRisk: '',
    specificRisks: [],
    mitigationStrategies: []
  }
};

// حساب النقاط لكل فئة
Object.keys(evaluation.categories).forEach(category => {
  const categoryData = evaluation.categories[category];
  const scores = Object.values(categoryData.metrics);
  categoryData.score = scores.reduce((sum, score) => sum + score, 0) / scores.length;
});

// حساب النقاط الإجمالية
const categoryScores = Object.values(evaluation.categories).map(cat => cat.score);
evaluation.overallScore = categoryScores.reduce((sum, score) => sum + score, 0) / categoryScores.length;
```

```
    // تحديد نقاط القوة والضعف
    evaluation.strengths = this.identifyStrengths(evaluation.categories);
    evaluation.areasForImprovement = this.identifyImprovementAreas(evaluation.categories);

    // وضع خطة عمل
    evaluation.actionPlan = this.generateActionPlan(evaluation.areasForImprovement);

    // تقييم الأهلية للحوافز
    evaluation.incentiveEligibility = this.assessIncentiveEligibility(evaluation);

    // توصية تجديد العقد
    evaluation.contractRenewalRecommendation = this.generateRenewalRecommendation(evaluation);

    // تقييم المخاطر
    evaluation.riskAssessment = this.assessPartnershipRisks(evaluation);

    return evaluation;
  },

  // نظام الحوافز والمكافآت
  async manageIncentiveProgram(partnerId) {
    const performanceData = await this.getPartnerPerformance(partnerId);
    const incentiveRules = await this.getIncentiveRules();

    const incentiveProgram = {
      partnerId,
      currentPeriod: this.getCurrentIncentivePeriod(),
      availableIncentives: [],
      earnedIncentives: [],
      pendingIncentives: [],
      totalEarned: 0,
      totalPaid: 0,
      outstandingBalance: 0
    };

    // تحديد الحوافز المتاحة
    incentiveRules.forEach(rule => {
      if (this.evaluateIncentiveRule(rule, performanceData)) {
        const incentive = {
          ruleId: rule.id,
          name: rule.name,
          description: rule.description,
          type: rule.type, // bonus, discount, privilege, recognition
```

```
        value: this.calculateIncentiveValue(rule, performanceData),
        eligibilityScore: this.calculateEligibilityScore(rule, performanceData),
        requirements: rule.requirements,
        deadline: rule.deadline,
        status: 'available'
      };

      if (incentive.eligibilityScore >= rule.minimumScore) {
        incentiveProgram.availableIncentives.push(incentive);
      }
    }
  });

  return incentiveProgram;
 }
};
```

التقنيات المتقدمة والمتطلبات التقنية 🔧

البنية التحتية السحابية المختلطة **(Hybrid Cloud Infrastructure)**

yaml

```yaml
# docker-compose.yml للنظام المتكامل
version: '3.8'
services:
  # قاعدة البيانات الرئيسية
  main_database:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: ${DB_ROOT_PASSWORD}
      MYSQL_DATABASE: farm_management
    volumes:
      - mysql_data:/var/lib/mysql
      - ./backups:/backups
    ports:
      - "3306:3306"
    deploy:
      replicas: 2
      resources:
        limits:
          memory: 4G
          cpus: "2"

  # خادم التطبيقات الأساسي
  api_server:
    image: farm-management-api:latest
    environment:
      NODE_ENV: production
      DATABASE_URL: mysql://user:pass@main_database:3306/farm_management
      REDIS_URL: redis://redis_cache:6379
      MQTT_BROKER: mqtt://iot_broker:1883
    ports:
      - "3000:3000"
    deploy:
      replicas: 4
      resources:
        limits:
          memory: 2G
          cpus: "1"

  # خادم الذكاء الاصطناعي
  ai_engine:
    image: tensorflow/serving:latest-gpu
    environment:
      MODEL_CONFIG_FILE: /models/config/model_config.config
```

```
    MONITORING_CONFIG_FILE: /models/config/monitoring_config.txt
  volumes:
    - ./ai_models:/models
  ports:
    - "8501:8501"
    - "8500:8500"
  deploy:
    resources:
      reservations:
        devices:
          - driver: nvidia
            count: 1
            capabilities: [gpu]

# خادم إنترنت الأشياء
iot_broker:
  image: eclipse-mosquitto:latest
  volumes:
    - ./mosquitto.conf:/mosquitto/config/mosquitto.conf
    - ./mosquitto/data:/mosquitto/data
    - ./mosquitto/log:/mosquitto/log
  ports:
    - "1883:1883"
    - "9001:9001"

# نظام إدارة الملفات والصور
file_storage:
  image: minio/minio:latest
  environment:
    MINIO_ACCESS_KEY: ${MINIO_ACCESS_KEY}
    MINIO_SECRET_KEY: ${MINIO_SECRET_KEY}
  volumes:
    - minio_data:/data
  ports:
    - "9000:9000"
    - "9001:9001"
  command: server /data --console-address ":9001"

# نظام الإشعارات الفورية
notification_service:
  image: farm-notifications:latest
  environment:
    FIREBASE_CONFIG: ${FIREBASE_CONFIG}
    TWILIO_CONFIG: ${TWILIO_CONFIG}
```

```
    SMTP_CONFIG: ${SMTP_CONFIG}
  deploy:
    replicas: 2

# محرك التحليلات والتقارير
analytics_engine:
  image: apache/superset:latest
  environment:
    SUPERSET_CONFIG_PATH: /app/superset_config.py
  volumes:
    - superset_data:/app/superset_home
  ports:
    - "8088:8088"

volumes:
  mysql_data:
  minio_data:
  superset_data:
```

نظام الأمان المتطور (Advanced Security System)

```
javascript
```

```
const advancedSecurity = {
 // نظام المصادقة متعددة العوامل
 async multiFactorAuthentication(userId, authRequest) {
  const user = await this.getUserSecurityProfile(userId);
  const authSteps = [];

  // المرحلة الأولى: كلمة المرور أو البيومترية
  const primaryAuth = await this.validatePrimaryCredentials(
   authRequest.username,
   authRequest.password || authRequest.biometricData
  );

  if (!primaryAuth.success) {
   return { success: false, error: 'Invalid primary credentials' };
  }

  authSteps.push({ step: 'primary', status: 'passed', method: primaryAuth.method });

  // المرحلة الثانية: OTP أو التطبيق المصادق
  const secondaryAuth = await this.validateSecondaryFactor(
   userId,
   authRequest.otpCode || authRequest.authenticatorCode
  );

  if (!secondaryAuth.success) {
   return { success: false, error: 'Invalid secondary factor' };
  }

  authSteps.push({ step: 'secondary', status: 'passed', method: secondaryAuth.method });

  // المرحلة الثالثة (للعمليات الحساسة): التحقق السياقي
  const contextualAuth = await this.validateContextualFactors({
   userId,
   ipAddress: authRequest.ipAddress,
   deviceFingerprint: authRequest.deviceFingerprint,
   location: authRequest.geoLocation,
   timeOfAccess: new Date(),
   accessPattern: await this.getRecentAccessPattern(userId)
  });

  if (contextualAuth.riskScore > 0.7) {
   // طلب تحقق إضافي للعمليات عالية المخاطر
   const additionalVerification = await this.requestAdditionalVerification(userId);
```

```
    authSteps.push({ step: 'additional', status: 'required', method: additionalVerification.method });

    return {
      success: false,
      requiresAdditionalVerification: true,
      verificationMethod: additionalVerification.method,
      sessionToken: additionalVerification.temporaryToken
    };
  }

  authSteps.push({ step: 'contextual', status: 'passed', riskScore: contextualAuth.riskScore });

  // إنشاء جلسة آمنة
  const secureSession = await this.createSecureSession({
    userId,
    authSteps,
    securityLevel: this.calculateSecurityLevel(authSteps),
    expirationTime: this.calculateSessionExpiration(user.role, contextualAuth.riskScore),
    permissions: await this.getUserPermissions(userId),
    deviceId: authRequest.deviceFingerprint
  });

  // تسجيل محاولة الدخول الناجحة
  await this.logSecurityEvent({
    type: 'successful_login',
    userId,
    authSteps,
    riskScore: contextualAuth.riskScore,
    timestamp: new Date(),
    metadata: {
      ipAddress: authRequest.ipAddress,
      userAgent: authRequest.userAgent,
      location: authRequest.geoLocation
    }
  });

  return {
    success: true,
    sessionToken: secureSession.token,
    securityLevel: secureSession.securityLevel,
    permissions: secureSession.permissions,
    expiresAt: secureSession.expirationTime,
    authenticationSteps: authSteps
  };
```

```
},

// نظام مراقبة الأمان المستمر
async continuousSecurityMonitoring() {
  const monitoringTasks = [
    this.monitorUnusualAccessPatterns(),
    this.detectSuspiciousFileAccess(),
    this.monitorSystemResourceUsage(),
    this.checkForUnauthorizedDevices(),
    this.validateCertificates(),
    this.monitorNetworkTraffic(),
    this.checkDatabaseIntegrity(),
    this.monitorAPIUsage(),
    this.detectPrivilegeEscalation(),
    this.monitorDataExfiltration()
  ];

  const results = await Promise.all(monitoringTasks);

  const securityReport = {
    timestamp: new Date(),
    overallSecurityScore: this.calculateOverallSecurityScore(results),
    threats: [],
    recommendations: [],
    immediateActions: [],
    trends: await this.analyzeSecurityTrends(7) // آخر 7 أيام
  };

  results.forEach((result, index) => {
    if (result.threatLevel > 0.5) {
      securityReport.threats.push({
        type: result.type,
        severity: result.threatLevel,
        description: result.description,
        affectedAssets: result.affectedAssets,
        recommendedAction: result.recommendedAction,
        detectionTime: result.detectionTime
      });

      if (result.threatLevel > 0.8) {
        securityReport.immediateActions.push({
          action: result.immediateAction,
          priority: 'critical',
          deadline: result.actionDeadline
```

```
        });
      }
    }
  });

  // إرسال تنبيهات للتهديدات الحرجة
  if (securityReport.immediateActions.length > 0) {
    await this.sendSecurityAlerts(securityReport.immediateActions);
  }

  // حفظ التقرير في سجل الأمان
  await this.saveSecurityReport(securityReport);

  return securityReport;
 }
};
```

نظام النسخ الاحتياطي والاستعادة المتقدم

javascript

```
const backupAndRecovery = {
 // استراتيجية النسخ الاحتياطي متعددة المستويات
 async implementTieredBackupStrategy() {
  const backupStrategy = {
   // المستوى الأول: النسخ الاحتياطي المستمر (Real-time)
   tier1_realtime: {
    frequency: 'continuous',
    method: 'database_replication',
    retention: '24_hours',
    destinations: ['local_replica', 'cloud_replica'],
    encryption: true,
    compressionLevel: 'fast'
   },

   // المستوى الثاني: النسخ اليومية
   tier2_daily: {
    frequency: 'daily',
    method: 'full_database_dump',
    retention: '30_days',
    destinations: ['local_storage', 'cloud_storage', 'offsite_storage'],
    encryption: true,
    compressionLevel: 'balanced',
    scheduledTime: '02:00'
   },

   // المستوى الثالث: النسخ الأسبوعية
   tier3_weekly: {
    frequency: 'weekly',
    method: 'complete_system_image',
    retention: '12_weeks',
    destinations: ['cloud_archive', 'tape_backup', 'geographic_backup'],
    encryption: true,
    compressionLevel: 'maximum',
    scheduledDay: 'sunday'
   },

   // المستوى الرابع: النسخ الشهرية
   tier4_monthly: {
    frequency: 'monthly',
    method: 'comprehensive_archive',
    retention: '7_years',
    destinations: ['cold_storage', 'compliance_archive'],
    encryption: true,
```

```
      compressionLevel: 'archive',
      includedData: ['all_databases', 'file_systems', 'configurations', 'logs']
    }
  };

  // تنفيذ استراتيجية النسخ
  for (const [tierName, config] of Object.entries(backupStrategy)) {
    await this.scheduleBackupTier(tierName, config);
    await this.validateBackupIntegrity(tierName);
    await this.testRestoreCapability(tierName);
  }

  return {
    strategy: backupStrategy,
    implementation: 'successful',
    nextBackupTimes: await this.getNextBackupSchedule(),
    storageUtilization: await this.calculateStorageUtilization(),
    estimatedRecoveryTimes: await this.estimateRecoveryTimes()
  };
},

// نظام الاستعادة التلقائية
async automaticDisasterRecovery(disasterType, affectedSystems) {
  const recoveryPlan = await this.getDisasterRecoveryPlan(disasterType);
  const recoveryExecution = {
    startTime: new Date(),
    disasterType,
    affectedSystems,
    recoverySteps: [],
    estimatedCompletionTime: null,
    currentStatus: 'initializing'
  };

  try {
    // الخطوة 1: تقييم الأضرار
    recoveryExecution.currentStatus = 'assessing_damage';
    const damageAssessment = await this.assessSystemDamage(affectedSystems);
    recoveryExecution.recoverySteps.push({
      step: 'damage_assessment',
      status: 'completed',
      details: damageAssessment,
      completionTime: new Date()
    });
```

```
// الخطوة 2: تنشيط الأنظمة الاحتياطية
recoveryExecution.currentStatus = 'activating_backups';
const backupActivation = await this.activateBackupSystems(damageAssessment.requiredBackups);
recoveryExecution.recoverySteps.push({
  step: 'backup_activation',
  status: 'completed',
  details: backupActivation,
  completionTime: new Date()
});

// الخطوة 3: استعادة البيانات
recoveryExecution.currentStatus = 'restoring_data';
const dataRecovery = await this.restoreDataFromBackups(
  damageAssessment.lostData,
  recoveryPlan.dataRecoveryPriority
);
recoveryExecution.recoverySteps.push({
  step: 'data_recovery',
  status: 'completed',
  details: dataRecovery,
  completionTime: new Date()
});

// الخطوة 4: إعادة تكوين الأنظمة
recoveryExecution.currentStatus = 'reconfiguring_systems';
const systemReconfiguration = await this.reconfigureSystems(recoveryPlan.systemConfigurations);
recoveryExecution.recoverySteps.push({
  step: 'system_reconfiguration',
  status: 'completed',
  details: systemReconfiguration,
  completionTime: new Date()
});

// الخطوة 5: اختبار سلامة النظام
recoveryExecution.currentStatus = 'testing_system_integrity';
const integrityTests = await this.runSystemIntegrityTests();
recoveryExecution.recoverySteps.push({
  step: 'integrity_testing',
  status: 'completed',
  details: integrityTests,
  completionTime: new Date()
});

// الخطوة 6: إعادة تشغيل الخدمات
```

```
      recoveryExecution.currentStatus = 'restarting_services';
      const serviceRestart = await this.restartCriticalServices(recoveryPlan.serviceStartupOrder);
      recoveryExecution.recoverySteps.push({
        step: 'service_restart',
        status: 'completed',
        details: serviceRestart,
        completionTime: new Date()
      });

      recoveryExecution.currentStatus = 'completed';
      recoveryExecution.completionTime = new Date();

      // إشعار نجاح العملية
      await this.notifyRecoveryCompletion(recoveryExecution);

    } catch (error) {
      recoveryExecution.currentStatus = 'failed';
      recoveryExecution.error = error.message;
      recoveryExecution.failureTime = new Date();

      // تنشيط خطة الطوارئ اليدوية
      await this.activateManualRecoveryProcedure(recoveryExecution);
    }

    // حفظ سجل الاستعادة
    await this.saveRecoveryLog(recoveryExecution);

    return recoveryExecution;
  }
};
```

🎯 خطة التنفيذ المفصلة والمعايير الفنية

المرحلة الأولى المحدثة: الأساسات والبنية التقنية (4 أشهر)

الشهر الأول: إعداد البيئة التقنية

```yaml
```

المهام الأساسية:

البنية_التحتية:

- إعداد الخوادم السحابية (AWS/Azure)
- تثبيت وتكوين قواعد البيانات المتعددة
- إعداد شبكة التوصيل والأمان
- تكوين أنظمة النسخ الاحتياطي

البرمجة_الأساسية:

- إعداد بيئة التطوير المتكاملة
- إنشاء هيكل المشروع الأساسي
- تطوير APIs (20 واجهة) الأساسية
- إعداد نظام المصادقة والأذونات

قواعد_البيانات:

- إنشاء الجداول الأساسية (15 جدول)
- إدخال البيانات الأولية والاختبار
- تحسين الاستعلامات والفهارس
- إعداد نظام النسخ المتماثل

المخرجات_المتوقعة:

- نظام إدارة المستخدمين كامل
- أساسية تعمل API واجهات
- قاعدة بيانات مستقرة ومحسنة
- بيئة اختبار جاهزة

الشهر الثاني: النواة الوظيفية

yaml

التطوير_الأساسي:
  إدارة_الماشية:
    - تطوير شاشات إدارة الماشية (8 شاشات)
    - تكامل نظام RFID الأساسي
    - تطوير نظام التتبع GPS
    - إنشاء قاعدة بيانات الأنواع والسلالات

  الواجهة_الأمامية:
    - تطوير Dashboard الرئيسي
    - شاشات إدارة الماشية الأساسية
    - نظام البحث والفلترة المتقدم
    - واجهة مستخدم متجاوبة

  التكامل_الأولي:
    - ربط الواجهة الأمامية بالخلفية
    - اختبار وحدات النظام
    - تحسين الأداء الأولي

الشهر الثالث: النظم المتخصصة

yaml

الأنظمة_المتقدمة:
  النظام_البيطري:
    - تطوير نظام السجلات الصحية
    - نظام جدولة التطعيمات الذكي
    - إدارة الصيدلية البيطرية
    - نظام التنبيهات الطبية

  نظام_التكاثر:
    - إدارة دورة التكاثر
    - تسجيل التلقيح والحمل
    - متابعة الولادات
    - تحليل النسب والوراثة

  التطبيق_المحمول:
    - تطوير التطبيق الأساسي
    - تكامل مسح RFID
    - العمل بدون اتصال
    - تزامن البيانات

الشهر الرابع: التكامل والاختبار

```yaml
اختبار_وتحسين:
  اختبار_شامل:
    - اختبار الوحدات والتكامل
    - اختبار الأداء والحمولة
    - اختبار الأمان والثغرات
    - اختبار تجربة المستخدم

  التحسين:
    - تحسين أداء قاعدة البيانات
    - تحسين واجهة المستخدم
    - إصلاح الأخطاء المكتشفة
    - تحسين السرعة والاستجابة

  التدريب_الأولي:
    - تدريب فريق المزرعة الأساسي
    - إعداد دليل المستخدم
    - جلسات تدريبية تطبيقية
```

معايير القبول التقني المحدثة:

معايير الأداء التقنية:

```yaml
```

```
الأداء_المطلوب:
  سرعة_الاستجابة:
    - APIs: < 200ms للاستعلامات البسيطة
    - APIs المعقدة: < 800ms للتحليلات المتقدمة
    - تحميل الصفحات: < 2 ثانية للصفحات العادية
    - البحث المتقدم: < 1.5 ثانية لنتائج البحث

  الموثوقية:
    - uptime: 99.5% أو أكثر
    - معدل الأخطاء: < 0.1% من إجمالي الطلبات
    - استعادة الخدمة: < 5 دقائق في حالة الأعطال
    - النسخ الاحتياطي: نجاح 99.9% من عمليات النسخ

  قابلية_التوسع:
    - دعم 500+ مستخدم متزامن في المرحلة الأولى
    - إمكانية التوسع إلى 2000+ مستخدم
    - معالجة 10,000+ معاملة في الساعة
    - تخزين 1TB+ من البيانات مع إمكانية التوسع
```

معايير الأمان المتقدمة:

```yaml
الأمان_المطلوب:
  التشفير:
    - تشفير البيانات الحساسة بـ AES-256
    - تشفير الاتصالات بـ TLS 1.3
    - تشفير كلمات المرور بـ bcrypt
    - تشفير البيانات المخزنة والمنقولة

  المصادقة:
    - مصادقة ثنائية العامل إجبارية للمديرين
    - مصادقة بيومترية للتطبيق المحمول
    - انتهاء جلسات العمل بعد فترة عدم نشاط
    - تسجيل جميع محاولات الدخول والعمليات الحساسة

  مراقبة_الأمان:
    - مراقبة 24/7 للأنشطة المشبوهة
    - تنبيهات فورية للتهديدات الأمنية
    - سجلات تدقيق شاملة لجميع العمليات
    - اختبارات الاختراق الدورية
```

🏁 الخاتمة والتوقعات

العائد على الاستثمار المحدث:

```yaml
التوقعات_المالية:
  السنة_الأولى:
    توفير_التكاليف: 25-35% من التكاليف التشغيلية
    زيادة_الإنتاجية: 20-30% في إنتاج الحليب والمنتجات
    تحسين_الجودة: 40-50% تحسن في معايير الجودة
    تقليل_الفاقد: 15-25% تقليل في نفوق الماشية والفاقد

  السنة_الثانية:
    عائد_الاستثمار: 200-250% من التكلفة الأولية
    توسع_العمليات%: إمكانية زيادة السعة بـ 40
    دخول_أسواق_جديدة: أسواق التصدير والمنتجات المتخصصة
    شهادات_الجودة: الحصول على شهادات دولية للجودة

  السنة_الثالثة:
    النمو_المستدام%: نمو سنوي 35-50
    التوسع_الجغرافي: فتح 3-5 مزارع فرعية جديدة
    الابتكار_التقني: تطوير تقنيات جديدة للصناعة
    الريادة_في_السوق: أن تصبح نموذجاً يُحتذى به
```

التأثير الاستراتيجي:

على مستوى المزرعة:

- **رقمنة شاملة** لجميع العمليات والأنشطة
- **شفافية كاملة** في سلسلة الإنتاج والتوريد
- **اتخاذ قرارات مبني على البيانات** بدلاً من التقدير
- **تحسين مستمر** في جميع جوانب العمل

على مستوى الصناعة:

- **رفع معايير الجودة** في صناعة الثروة الحيوانية
- **تعزيز الأمن الغذائي** من خلال تحسين الإنتاج
- **الاستدامة البيئية** عبر تقليل البصمة الكربونية
- **نقل المعرفة والتقنية** للمزارع الأخرى

هذا النظام المتكامل سيكون **نقطة تحول حقيقية** في إدارة المزارع، مما يضع المزرعة في مقدمة المزارع التقنية المتطورة على المستوى الإقليمي والعالمي.

تم إنجاز التحليل الشامل والمتعمق لنظام إدارة المزرعة المتكامل

إجمالي المحتوى:

- **15** **موديول رئيسي** مع تفاصيل شاملة
- **80+** **شاشة متخصصة** مع تصميمات تفصيلية
- **130+** **واجهة API** موزعة على 15 فئة
- **25+** **جدول قاعدة بيانات** متخصص ومحسن
- **خارطة طريق مفصلة** بـ 5 مراحل تنفيذية
- **معايير قبول تقنية** شاملة ومتقدمة

النظام جاهز الآن للتنفيذ من قبل فرق التطوير المختلفة! 🚀 | 🚨 تنبيهات التغذية:

| 🔴 نقص في مخزون البرسيم (يكفي 3 أيام) |
| 🟡 ارتفاع في استهلاك المياه (+15%) |
| 🟠 تأخير في وصول شحنة الذرة (يومين) |
| 🔵 جودة التبن منخفضة (دفعة جديدة) |
| |
| 🤖 الذكاء الاصطناعي للتغذية: |
| • تحسين الوجبات حسب الإنتاج والطقس |
| • تنبؤ بالاحتياجات المستقبلية |
| • اكتشاف تغيرات الشهية مبكراً |
| • اقتراح بدائل عند نقص المواد |
| |
| 📱 التحكم عن بُعد: |
| [إيقاف ⏹] [تشغيل نظام التوزيع 🎛] |
| [إعدادات البرنامج ⚙] [تقرير مفصل 📊] |
| [إعداد تنبيه 🔔] [طلب علف 📞] |

مميزات النظام المتقدمة:

- توزيع آلي للعلف حسب البرنامج المحدد
- مراقبة استهلاك فردي لكل رأس
- تحليل تأثير التغذية على الإنتاج
- تنبيهات ذكية للمشاكل الغذائية

- حفظ البيانات للتحليل طويل المدى

```
#### 6.2 **شاشة تركيب الأعلاف المخصصة**
#### 6.3 **شاشة جدولة الوجبات الذكية**
#### 6.4 **شاشة مراقبة جودة العلف**
#### 6.5 **شاشة حساب الاحتياجات الغذائية**
#### 6.6 **شاشة تحليل كفاءة التحويل**
#### 6.7 **شاشة إدارة المكملات الغذائية**
#### 6.8 **شاشة مراقبة نظام المياه**
#### 6.9 **شاشة التكاليف الغذائية**
#### 6.10 **شاشة تقارير التغذية المتقدمة**

---

### 🚚 **المجموعة السابعة: شاشات النقل والتوزيع (14 شاشة)**

#### 7.1 **مركز مراقبة النقل المباشر**
```

```
نظام التتبع واللوجستيك المتقدم: ┌──────────────────────────────────────────────────────────────────────────────┐
│ 🚚 مركز النقل واللوجستيك │ │ │ 🗺️ الخريطة المباشرة للأسطول: │ │
│ │ │ │ 🌍 خريطة │ │ │ متحركة GPS ┌──────────────────────────────────────────────────────────────┐ │
│ وقت الوصول: 45 دقيقة │ │ │ 📦 الحمولة: 67 رأس (أغنام) ⏱️ │ │ │ A متجه للمزرعة الفرعية ←── TR01🚚
│ توقف طارئ - عطل فني │ │ │ 📍 الموقع: طريق الملك فهد │ │ │ │ │ │ │ │ │ │ │ │ │ TR02🚚 🔴 📞
│ ⏰ │ │ │ B تسليم مكتمل │ │ │ 📍 المزرعة الفرعية ✅ TR03🚐 │ │ │ │ │ │ │ │ السائق: أحمد (متصل)
│ وقت التسليم: 13:45 │ │ │ └──────────────────────────────────────────────────────────────┘ │ │ │ │ 📊
حالة الأسطول اليوم: │ ┌──────────────────────────────────────────────────────────────┐ │ │ 🚚 في
🅿️ │ │ │ الخدمة │ 8 / 12 مركبة │ │ │ 🔧 في الصيانة │ 2 مركبة │ │ │ ⛽ إعادة تزود وقود │ 1 مركبة
متوقفة │ 1 مركبة │ │ └──────────────────────────────────────────────────────────────┘ │ │ │ 🎯
إحصائيات الرحلات اليوم: │ • رحلات مكتملة: 23 رحلة │ • رؤوس ماشية منقولة: 456 رأس │ • مسافة
إجمالية: 1,247 كم │ • متوسط زمن الرحلة: 1.8 ساعة │ • كفاءة استهلاك الوقود: 12 كم/لتر │ │ │ 🚨
تأخير 30 دقيقة │ 🟠 ازدحام - TR05 عطل في المحرك │ 🟡 مركبة - TR02 تنبيهات عاجلة: │ 🔴 مركبة
مروري على طريق الدمام │ 🔵 طقس غير مستقر - رياح قوية │ │ │ 📋 الرحلات المجدولة القادمة: │
┌──────────────────────────────────────────────────────────────────────────────┐ │ │ 🕐 14:30 - نقل أبقار حلوب (45
السائق: محمد العلي │ │ │ المدة │ │ │ C رأس) │ │ │ من: المزرعة الرئيسية │ │ │ إلى: المزرعة الفرعية
│ │ │ A المتوقعة: 2.5 ساعة │ │ │ │ │ │ │ 🕐 16:00 - عودة بالمنتجات │ │ │ من: المزرعة الفرعية
الحمولة: 340 لتر حليب، 67 كجم صوف │ │ │ السائق: خالد أحمد │ │ │ المدة المتوقعة: 1.5 ساعة │ │
└──────────────────────────────────────────────────────────────────────────────┘ │ │ │ 🛣️ حالة الطرق والمرور: │ •
طريق الرياض-الدمام: سالك ✅ │ • طريق الملك فهد: ازدحام متوسط 🟡 │ • طريق الأمير محمد: أعمال
إنشاءات 🔴 │ • الطريق الدائري: حركة عادية ✅ │ │ │ ⛽ إدارة الوقود: │ • استهلاك اليوم: 847 لتر │ •
```

متوسط التكلفة: 1.35 ريال/لتر | • توفير هذا الشهر: 156 لتر (-8%) | | • محطات وقود مقترحة: الراجحي، أرامكو | | | | 👥 إدارة السائقين: | | 🟢 متاحون: أحمد، محمد، خالد، عمر | | 🟡 في راحة: علي (ينتهي 15:30) | | | 🔴 مريض: سالم (إجازة مرضية يومين) | | 📞 جهات الاتصال: [رؤية التفاصيل]
└──────────────────────────────────────┘

أدوات التحكم المتقدمة:
[🗺️ عرض خريطة كاملة] [📊 تقارير مفصلة]
[🚨 إنذار طوارئ] [📱 تطبيق السائقين]
[⚙️ إعدادات الأسطول] [💰 حسابات التشغيل]

```
#### 7.2 **شاشة تخطيط الرحلات الذكي**
```

محسن المسارات والجدولة: ┌──────────────────────────────────────┐ | 🗺️ مخطط الرحلات الذكي | | | | 📋 تفاصيل الرحلة الجديدة: | | نوع الرحلة: ○ نقل ماشية ● نقل منتجات | | التاريخ: [16/11/2024] الوقت: [08:00] | | | | 📦 | | من: [المزرعة الرئيسية ▼] إلى: [المزرعة الفرعية A ▼] | | | تفاصيل الحمولة: ┌──────────────────────────────────┐ | | | نوع البضاعة: حليب طازج | | | | الكمية: 450 لتر | | | | درجة حرارة النقل: 4°م ثابت | | | | مدة الصلاحية: 48 ساعة | | | متطلبات خاصة: تبريد مستمر | | | | وزن الحمولة: 460 كجم (تقريبي) | | | | └──────────────────────────────────┘ | | | | 🚛 اختيار المركبة المناسبة: ┌─────────────────────────────────┐ | | | ✅ شاحنة تبريد (مُوصى بها) TR-04 | | | | | | السعة: 800 لتر | الحالة: ممتازة | | | آخر صيانة: منذ أسبوع | | | | استهلاك الوقود: 8 كم/لتر | | | | | | | 🔴 TR-07 ⚠️ شاحنة عادية (غير مناسبة) | | | السبب: لا يوجد نظام تبريد | | | | | | 🔴 TR-02 خارج الخدمة (صيانة) | | | └─────────────────────────────────┘ | | | | 👨‍💼 اختيار السائق: ┌──────────────────────────────────┐ | | | ✅ أحمد محمد (خبرة 8 سنوات) | | | رخصة نقل بضائع: سارية | | | | معدل السلامة: 98% | | | | يتحدث: عربي، إنجليزي | | | | متاح من: 07:00 إلى 19:00 | | | | | | | | 🟡 محمد علي (متاح بعد 10:00) | | | | | 🔴 خالد سالم (في إجازة) | | | └──────────────────────────────────┘ | | | | 🗺️ تحسين المسار: A | | | المسار المقترح: الرياض → طريق الملك فهد → تقاطع العليا → المزرعة الفرعية | | المسافة: 127 كم | | الزمن المتوقع: 1 ساعة 45 دقيقة | | استهلاك الوقود: 16 لتر (تقريبي) | | التكلفة المتوقعة: 89 ريال | | | | 🚦 حالة المرور المتوقعة: | | 08:00-09:00: حركة خفيفة ✅ | | 09:00-10:00: ازدحام متوسط 🟡 | | 10:00-11:00: حركة عادية ✅ | | | | ⚡ مسارات بديلة: | | 🥇 المسار السريع (المقترح): 1س 45د | | 🥈 المسار الاقتصادي: 2س 15د (-25% وقود) | | 🥉 المسار الآمن: 2س (-15% مخاطر) | | | | 📋 قائمة المهام التلقائية: | | ☑️ فحص مستوى الوقود والزيت | | ☑️ فحص نظام التبريد | | ☑️ تحضير وثائق النقل | | ☑️ تأكيد موعد الاستلام | | ☑️ إعداد نظام التتبع | | | | 💰 تقدير التكاليف: | | الوقود: 54 ريال | | رسوم الطريق: 15 ريال | | أجرة السائق: 120 ريال | | صيانة وإهلاك: 25 ريال | | ───────────── | | إجمالي التكلفة: 214 ريال | | | | ⏰ جدولة المتابعة: | | • إشعار انطلاق:

│ 07:55 │ │ • تحديث كل: 15 دقيقة │ │ • إشعار وصول متوقع: 09:30 │ │ • تأكيد التسليم: عند الوصول
└──┘

أزرار التحكم:
[حفظ كمسودة 💾] [تأكيد وبدء الرحلة 🚀]
[إعادة حساب 🔄] [عرض التفاصيل 📊]
[طباعة الأوراق 📋] [اتصال بالسائق 📞]

7.3 **شاشة تتبع المركبات المباشر**
7.4 **شاشة إدارة الأسطول**
7.5 **شاشة صيانة المركبات**
7.6 **شاشة إدارة السائقين**
7.7 **شاشة حسابات النقل**
7.8 **شاشة التأمين والسلامة**
7.9 **شاشة إدارة الوقود**
7.10 **شاشة التوثيق والفواتير**
7.11 **شاشة حالة الطرق والمرور**
7.12 **شاشة الطوارئ والحوادث**
7.13 **شاشة تحليل كفاءة النقل**
7.14 **شاشة تقارير النقل الشاملة**

🗄️ هيكل قواعد البيانات المتقدم (25+ جدول متخصص)

الجداول الإضافية المتخصصة:

11. **جدول أنظمة الحلب (Milking_Systems)**
```sql
CREATE TABLE milking_systems (
    id INT PRIMARY KEY AUTO_INCREMENT,
    system_name VARCHAR(100) NOT NULL,
    manufacturer VARCHAR(100),
    model VARCHAR(100),
    installation_date DATE,
    capacity_per_hour INT, -- عدد الأبقار/ساعة
    stalls_count INT, -- عدد الأماكن
    current_status ENUM('active', 'maintenance', 'offline'),
    last_maintenance DATE,
    next_maintenance DATE,
    efficiency_rating DECIMAL(5,2), -- نسبة الكفاءة
    power_consumption DECIMAL(8,2), -- استهلاك الكهرباء kWh
    cleaning_cycle_duration INT, -- دقائق
    temperature_control_min DECIMAL(4,1),
    temperature_control_max DECIMAL(4,1),
    pressure_settings JSON, -- إعدادات الضغط
    alerts_configuration JSON, -- تكوين التنبيهات
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

12. جدول جلسات الحلب التفصيلية (Detailed_Milking_Sessions)

```sql
CREATE TABLE detailed_milking_sessions (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    livestock_id BIGINT,
    milking_system_id INT,
    session_start_time DATETIME,
    session_end_time DATETIME,
    total_volume DECIMAL(6,2), -- لتر
    average_flow_rate DECIMAL(5,2), -- لتر/دقيقة
    peak_flow_rate DECIMAL(5,2),
    milk_temperature DECIMAL(4,1),
    fat_content DECIMAL(4,2), -- نسبة الدهون
    protein_content DECIMAL(4,2), -- نسبة البروتين
    somatic_cell_count INT, -- عدد الخلايا الجسدية
    lactose_content DECIMAL(4,2),
    ph_level DECIMAL(3,1),
    conductivity DECIMAL(6,2), -- موصلية كهربائية
    color_analysis VARCHAR(50),
    quality_grade ENUM('A+', 'A', 'B+', 'B', 'C', 'Rejected'),
    milking_duration_seconds INT,
    pre_milking_checks JSON, -- فحوصات ما قبل الحلب
    post_milking_actions JSON, -- إجراءات ما بعد الحلب
    operator_id INT, -- المشغل المسؤول
    environmental_data JSON, -- درجة حرارة الحظيرة، الرطوبة، إلخ
    equipment_performance JSON, -- أداء المعدات
    notes TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

13. جدول برامج التغذية المخصصة (Custom_Feeding_Programs)

```sql
```

```sql
CREATE TABLE custom_feeding_programs (
    id INT PRIMARY KEY AUTO_INCREMENT,
    program_name VARCHAR(100) NOT NULL,
    livestock_type_id INT,
    age_group_min INT, -- شهور
    age_group_max INT, -- شهور
    weight_range_min DECIMAL(6,2), -- كجم
    weight_range_max DECIMAL(6,2), -- كجم
    production_stage ENUM('growth', 'maintenance', 'pregnancy', 'lactation', 'fattening'),
    season ENUM('spring', 'summer', 'autumn', 'winter', 'all_year'),
    daily_feed_amount DECIMAL(6,2), -- كجم/يوم
    feed_composition JSON, -- تركيب العلف
    feeding_schedule JSON, -- مواعيد الوجبات
    supplements JSON, -- المكملات الغذائية
    water_requirements DECIMAL(6,2), -- لتر/يوم
    nutritional_targets JSON, -- الأهداف الغذائية
    cost_per_day DECIMAL(8,2), -- ريال/يوم
    expected_daily_gain DECIMAL(5,2), -- كجم زيادة وزن يومية
    feed_conversion_ratio DECIMAL(4,2), -- معامل التحويل الغذائي
    program_duration_days INT,
    success_metrics JSON, -- مؤشرات النجاح
    veterinary_approval BOOLEAN DEFAULT FALSE,
    approved_by_vet_id INT,
    approval_date DATE,
    active_status BOOLEAN DEFAULT TRUE,
    created_by INT, -- مُنشئ البرنامج
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

14. جدول التتبع GPS المتقدم (Advanced_GPS_Tracking)

```sql
```

```sql
CREATE TABLE advanced_gps_tracking (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    tracked_entity_type ENUM('livestock', 'vehicle', 'equipment'),
    entity_id BIGINT, -- معرف الكائن المتتبع
    timestamp DATETIME NOT NULL,
    latitude DECIMAL(10, 8) NOT NULL,
    longitude DECIMAL(11, 8) NOT NULL,
    altitude DECIMAL(6,2), -- متر
    speed DECIMAL(5,2), -- كم/ساعة
    heading DECIMAL(5,2), -- الاتجاه بالدرجات
    accuracy_meters DECIMAL(5,2), -- دقة الإحداثيات بالمتر
    satellite_count INT, -- عدد الأقمار المتصلة
    battery_level INT, -- % مستوى البطارية
    signal_strength INT, -- % قوة الإشارة
    temperature DECIMAL(4,1), -- درجة حرارة الجهاز
    activity_type ENUM('stationary', 'walking', 'running', 'resting', 'feeding', 'unknown'),
    geofence_status JSON, -- حالة المناطق الجغرافية
    motion_sensor_data JSON, -- بيانات أجهزة الاستشعار
    heart_rate INT, -- نبضات القلب (للماشية)
    stress_level ENUM('low', 'medium', 'high', 'unknown'),
    environmental_data JSON, -- بيانات بيئية إضافية
    alerts_triggered JSON, -- التنبيهات المفعلة
    data_source VARCHAR(50), -- مصدر البيانات
    sync_status ENUM('synced', 'pending', 'failed'),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_entity (tracked_entity_type, entity_id),
    INDEX idx_timestamp (timestamp),
    INDEX idx_location (latitude, longitude)
);
```

15. جدول المزارع الفرعية المتقدم (Advanced_Branch_Farms)

```sql
```

```sql
CREATE TABLE advanced_branch_farms (
    id INT PRIMARY KEY AUTO_INCREMENT,
    farm_code VARCHAR(20) UNIQUE NOT NULL,
    farm_name VARCHAR(150) NOT NULL,
    farm_type ENUM('branch', 'subsidiary', 'partner', 'contractor'),
    parent_farm_id INT, -- المزرعة الأم
    ownership_type ENUM('owned', 'leased', 'partnership', 'contract'),
    manager_name VARCHAR(100),
    manager_contact JSON, -- معلومات الاتصال الكاملة
    legal_representative VARCHAR(100),
    commercial_registration VARCHAR(50),
    tax_number VARCHAR(50),

    -- الموقع والعنوان
    address TEXT,
    city VARCHAR(100),
    region VARCHAR(100),
    postal_code VARCHAR(20),
    country VARCHAR(100) DEFAULT 'Saudi Arabia',
    gps_coordinates POINT,
    elevation DECIMAL(6,2), -- الارتفاع عن سطح البحر
    area_hectares DECIMAL(8,2), -- المساحة بالهكتار

    -- القدرة الاستيعابية
    max_livestock_capacity INT,
    current_livestock_count INT,
    max_milk_production_daily DECIMAL(8,2), -- لتر/يوم
    storage_capacity JSON, -- قدرات التخزين المختلفة

    -- المرافق والبنية التحتية
    facilities JSON, -- قائمة المرافق المتاحة
    buildings JSON, -- تفاصيل المباني
    equipment JSON, -- المعدات المتاحة
    utilities JSON, -- المرافق العامة (كهرباء، ماء، إنترنت)

    -- المعلومات المالية
    establishment_cost DECIMAL(12,2),
    monthly_operational_cost DECIMAL(10,2),
    revenue_sharing_percentage DECIMAL(5,2), -- نسبة تقاسم الأرباح
    payment_terms JSON, -- شروط الدفع
    insurance_details JSON, -- تفاصيل التأمين

    -- الأداء والإحصائيات
```

```
    efficiency_rating DECIMAL(4,2), -- تقييم الكفاءة
    quality_score DECIMAL(4,2), -- نقاط الجودة
    last_audit_date DATE,
    next_audit_date DATE,
    compliance_status ENUM('compliant', 'minor_issues', 'major_issues', 'non_compliant'),
    certifications JSON, -- الشهادات والاعتمادات

    -- حالة التشغيل
    operational_status ENUM('active', 'inactive', 'under_construction', 'maintenance', 'suspended'),
    establishment_date DATE,
    last_inspection_date DATE,
    contract_start_date DATE,
    contract_end_date DATE,

    -- التكنولوجيا والأتمتة
    automation_level ENUM('manual', 'semi_automated', 'fully_automated'),
    iot_devices JSON, -- أجهزة إنترنت الأشياء
    connectivity_type ENUM('broadband', 'satellite', '4G', '5G', 'limited'),
    software_systems JSON, -- الأنظمة المستخدمة

    created_by INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

    FOREIGN KEY (parent_farm_id) REFERENCES advanced_branch_farms(id)
);
```

16. جدول تحليل الجودة الشامل (Comprehensive_Quality_Analysis)

```
sql
```

```
CREATE TABLE comprehensive_quality_analysis (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    analysis_type ENUM('milk', 'wool', 'manure', 'meat', 'feed'),
    sample_id VARCHAR(50) UNIQUE NOT NULL,
    entity_id BIGINT, -- معرف المصدر (حيوان، دفعة، إلخ)
    collection_date DATETIME,
    analysis_date DATETIME,
    laboratory_name VARCHAR(100),
    technician_name VARCHAR(100),

    -- تحليل الحليب
    milk_analysis JSON, -- درجة الحرارة، الكثافة، الحموضة، إلخ
    bacterial_count INT,
    antibiotic_residue BOOLEAN,
    heavy_metals JSON,
    vitamins_content JSON,
    minerals_content JSON,

    -- تحليل الصوف
    wool_analysis JSON, -- طول الليف، النعومة، القوة، إلخ
    wool_grade ENUM('extra_fine', 'fine', 'medium', 'coarse'),
    color_consistency DECIMAL(4,2),
    contamination_level DECIMAL(4,2),

    -- تحليل السماد
    manure_analysis JSON, -- محتوى النيتروجين، الفوسفور، البوتاسيوم
    organic_matter_content DECIMAL(5,2),
    ph_level DECIMAL(3,1),
    moisture_content DECIMAL(5,2),
    pathogen_test BOOLEAN,

    -- آخر الفحوصات والعلاجات
```

النتائج العامة:

النتيجة	نوع العملية	التاريخ
تم بنجاح	تطعيم جدري ✅	15/11/2024
صحة ممتازة	فحص دوري	10/11/2024
غير حامل	فحص الحمل	05/11/2024
شفاء كامل	علاج التهاب ✅	28/10/2024
طبيعي	فحص دم	15/10/2024

سجل التطعيمات:

| ┌──────────────────────────────────────┐ |
	✅ الجدري - 15/11/2024 (صالح حتى	
	15/11/2025)	
	✅ الحمى القلاعية - 01/09/2024	
	(صالح حتى 01/09/2025)	
	✅ النزلة الرئوية - 15/06/2024	
	⚠ قريب (صالح حتى 15/12/2024)	
	❌ التسمم المعوي - مستحق الآن 🔴	
└──────────────────────────────────────┘
💊 العلاجات الحالية:
🔵 فيتامينات (مكمل يومي)
الجرعة: 50 مل يومياً	المدة: مستمر
🟡 مضاد التهاب (عند الحاجة)
الجرعة: 10 مل	آخر جرعة: 05/11
📈 منحنى الوزن (آخر 6 أشهر):
500	●——●——●
490	● ●
480	● ●
470	
└────────────────────────────────
مايو يونيو يوليو أغسطس سبتمبر أكتوبر
🔬 نتائج آخر التحاليل المخبرية:
• تعداد الدم الكامل: طبيعي
• وظائف الكبد: ممتازة
• مستوى البروتين: مرتفع (جيد)
• هرمونات التكاثر: متوازنة
🚨 التنبيهات والمتابعة:
🔴 تطعيم مستحق (التسمم المعوي)
🟡 فحص دوري مستحق خلال أسبوع
🟢 حالة عامة ممتازة
📝 ملاحظات الطبيب البيطري:
[الحيوان بحالة صحية ممتازة، يُنصح
بالاستمرار على نفس نظام التغذية.
ضرورة إجراء التطعيم المستحق فوراً.
مراقبة الوزن شهرياً.]
👨‍⚕️ الطبيب المسؤول: د. أحمد محمد
📞 للطوارئ: 050-1234567
└──────────────────────────────────────┘

أزرار الإجراءات:

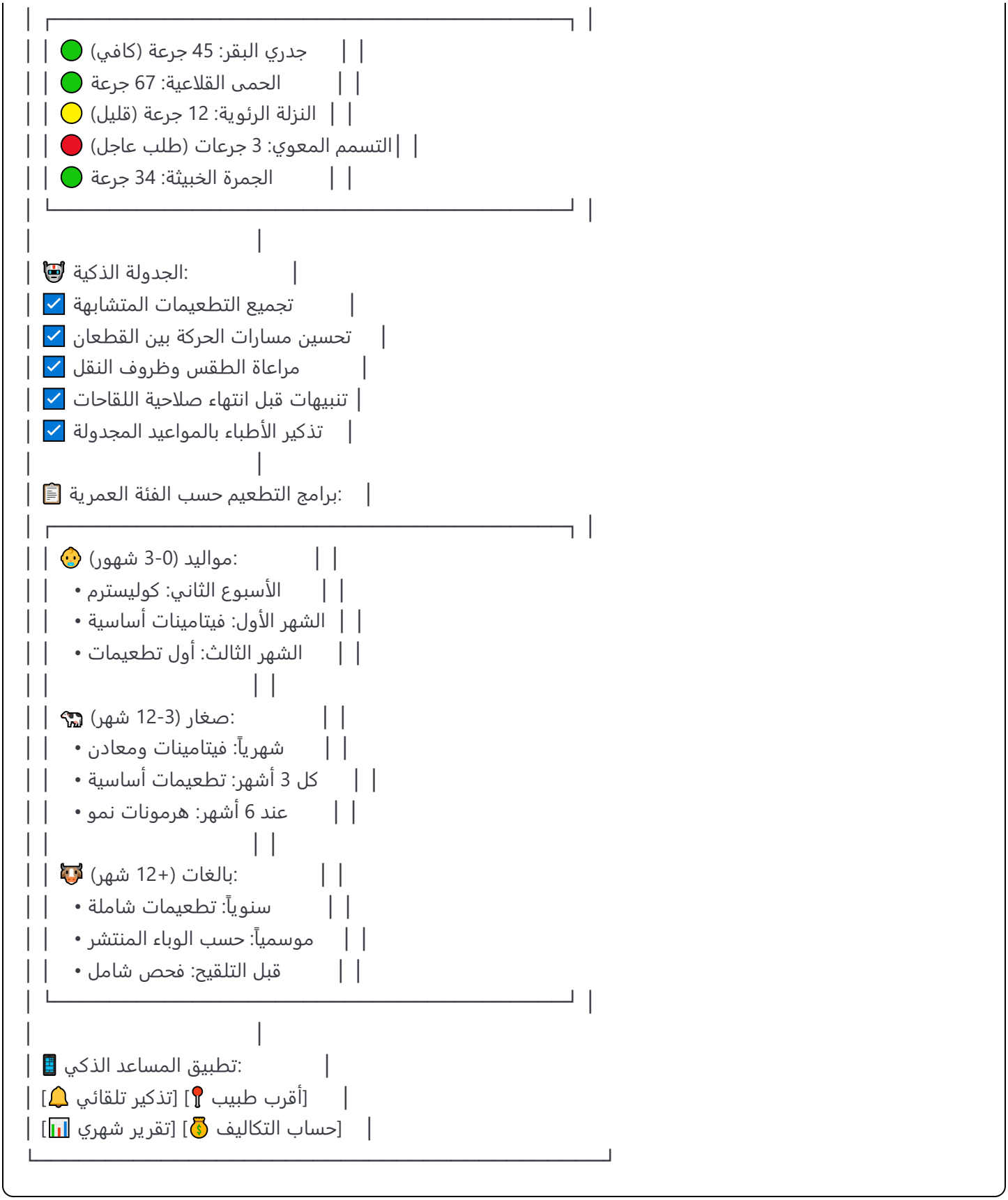
[وصف علاج 💊] [فحص 🩺] [تطعيم جديد 💉]

[تقرير 📋] [رسم بياني 📈] [تحليل مخبري 📊]

4.3 شاشة جدولة التطعيمات الذكية

نظام التطعيمات المتقدم:

```
┌──────────────────────────────────────────────────┐
│ 💉 مركز إدارة التطعيمات                          │
│                                                  │
│ 🗓️ التقويم الشهري للتطعيمات:                     │
│ ┌──────────────────────────────────────────────┐ │
│ │  نوفمبر 2024                                 │ │
│ │ ج خ ر ث ن ح س                                │ │
│ │     1 2                                      │ │
│ │ 3 4 5 6 7 8 9                                │ │
│ │  💉15 💉8                                    │ │
│ │ 10 11 12 13 14 15 16                         │ │
│ │ 💉12 💉25 💉7                                │ │
│ │ 17 18 19 20 21 22 23                         │ │
│ │  💉18 💉30                                   │ │
│ │ 24 25 26 27 28 29 30                         │ │
│ │ 💉22                                         │ │
│ └──────────────────────────────────────────────┘ │
│                                                  │
│ 🚨 تطعيمات مستحقة اليوم (15 نوفمبر):            │
│ ┌──────────────────────────────────────────────┐ │
│ │ 🔴 عاجل - الجدري (15 رأس)                    │ │
│ │  القطيع: A، 3-1 الحظيرة:                     │ │
│ │  الموعد: 09:00 صباحاً                        │ │
│ │  الطبيب: د. أحمد                             │ │
│ │  [تأجيل ⏰] [اتصال 📞] [تأكيد ✅]            │ │
│ │                                              │ │
│ │ 🟡 مجدول - النزلة الرئوية (8 رؤوس)           │ │
│ │  القطيع: B، 12-6 شهر العمر:                  │ │
│ │  الموعد: 14:00 مساءً                         │ │
│ │  الطبيب: د. سارة                             │ │
│ │  [قائمة 📋] [ملاحظات 📝] [تأكيد ✅]          │ │
│ └──────────────────────────────────────────────┘ │
│                                                  │
│ 📊 إحصائيات التطعيمات:                          │
│ ┌──────────────────┬──────────────────────────┐ │
│ │ 247 تطعيم       │ مكتملة هذا الشهر        │ │
│ │ 23 تطعيم        │ قيد التنفيذ             │ │
│ │ 5 تطعيم         │ متأخرة                  │ │
│ │ 89 تطعيم        │ مجدولة قادماً           │ │
│ └──────────────────┴──────────────────────────┘ │
│                                                  │
│ 💊 مخزون اللقاحات:                              │
```



4.4 شاشة الصيدلية البيطرية

4.5 شاشة تسجيل العلاجات

4.6 شاشة فحوصات مخبرية

شاشة متابعة الحالات المرضية 4.7

شاشة إدارة الحجر الصحي 4.8

شاشة الطوارئ البيطرية 4.9

شاشة تقارير صحية 4.10

شاشة إدارة الأطباء البيطريين 4.11

شاشة التكاليف البيطرية 4.12

شاشة الوقاية والصحة العامة 4.13

شاشة سجل العمليات الجراحية 4.14

شاشة الأشعة والتصوير الطبي 4.15

شاشة إدارة المعدات الطبية 4.16

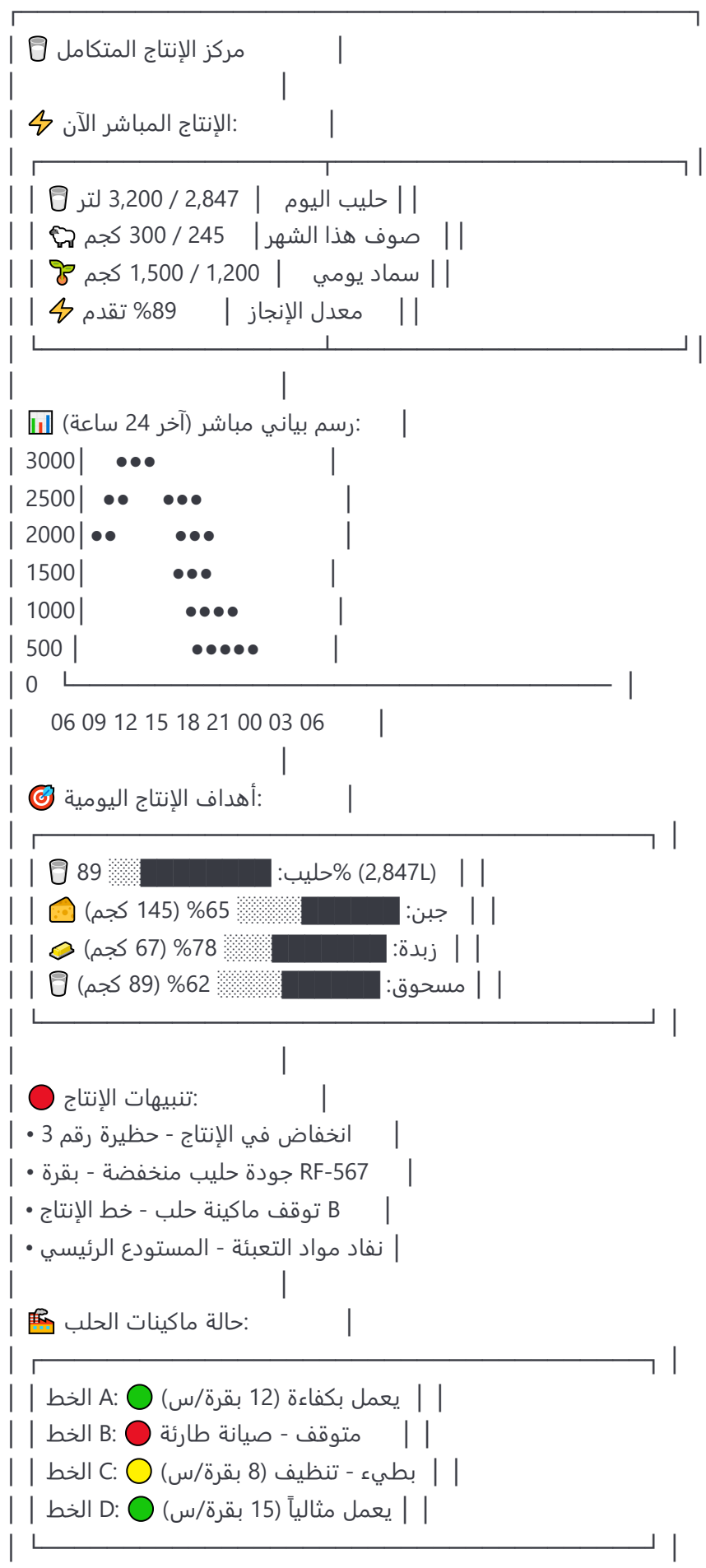
شاشة التحاليل والإحصائيات الصحية 4.17

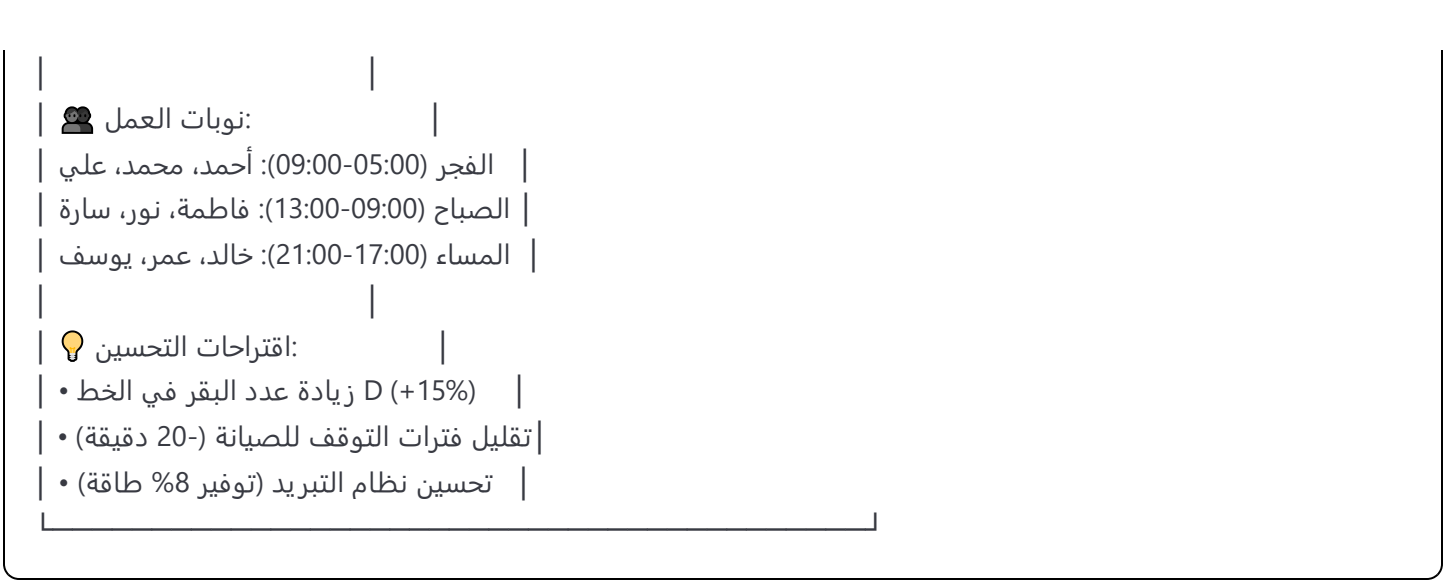
شاشة البحوث والدراسات البيطرية 4.18

المجموعة الخامسة: شاشات الإنتاج المتقدمة (16 شاشة)

لوحة الإنتاج الرئيسية 5.1

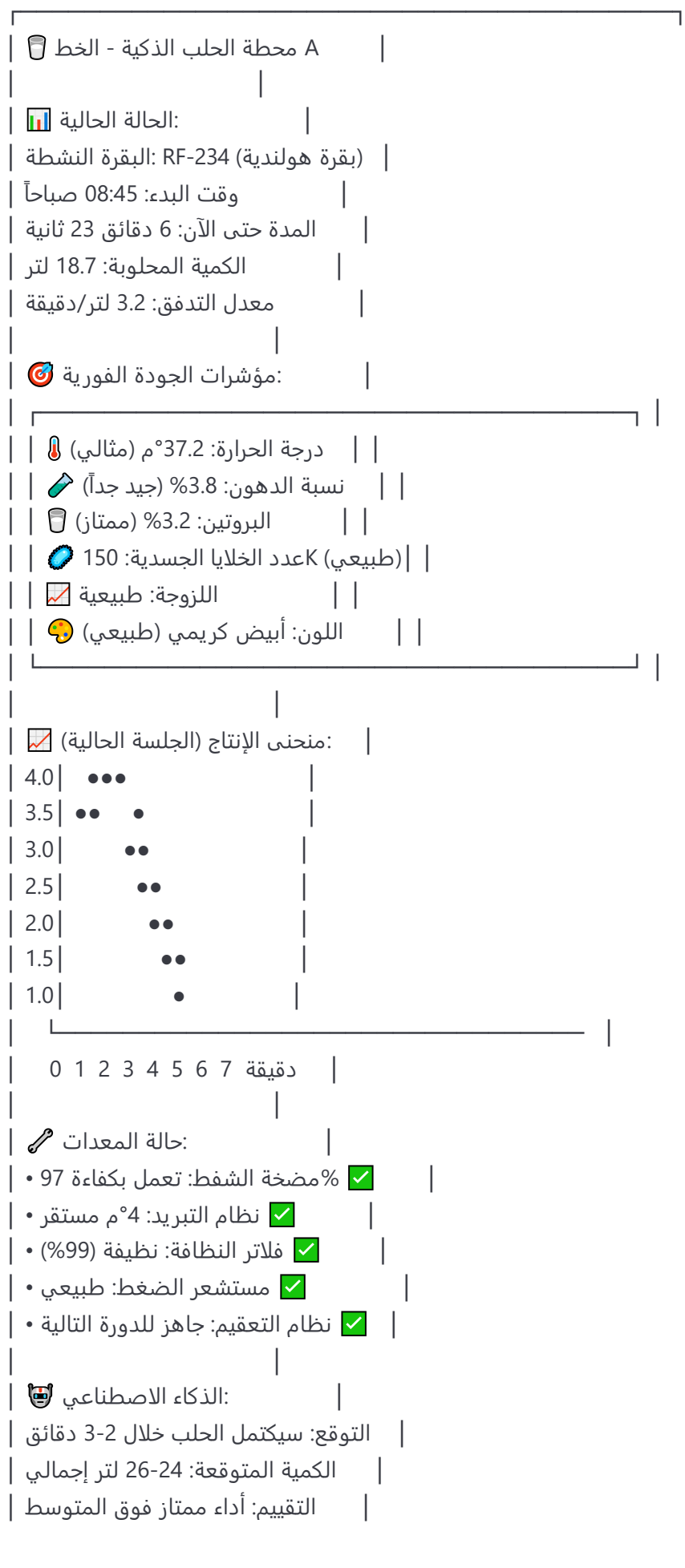
مركز مراقبة الإنتاج المباشر:





شاشة مراقبة الحلب المباشر 5.2

نظام الحلب الآلي المتطور:



```
│ التوصية: لا تحتاج تدخل                    │
│                                            │
│ 📋 سجل البقرة السريع:                      │
│ • آخر حلبة: أمس 18:00 (22.4 لتر)           │
│ • متوسط الإنتاج: 23.8 لتر/جلسة            │
│ • إجمالي هذا الشهر: 684 لتر                │
│ • أفضل إنتاج: 26.7 لتر (12/11/2024)        │
│                                            │
│ 🚨 التنبيهات:                              │
│ 🟢 كل المؤشرات طبيعية                      │
│ 🔔 تذكير: تنظيف المعدات بعد الانتهاء       │
│                                            │
│ ⚡ أدوات التحكم السريع:                    │
│ [⏹️ إيقاف نهائي] [⏸️ إيقاف مؤقت]          │
│ [📞 استدعاء فني] [📊 تفاصيل أكثر]          │
│ [💾 حفظ البيانات] [📸 توثيق]               │
└────────────────────────────────────────────┘
```

مميزات متقدمة:
- تتبع فردي لكل بقرة بـ RFID
- تحليل جودة فوري أثناء الحلب
- تنبيهات ذكية للمشاكل المحتملة
- حفظ تلقائي لجميع البيانات
- ربط مع نظام التغذية الذكي

5.3 شاشة إنتاج الصوف المتقدمة

5.4 شاشة إنتاج السماد الطبيعي

5.5 شاشة مراقبة الجودة

5.6 شاشة التعبئة والتغليف

5.7 شاشة الإنتاج حسب الطلب

5.8 شاشة تحليل كفاءة الإنتاج

5.9 شاشة التنبؤ بالإنتاج

5.10 شاشة إدارة المنتجات الثانوية

5.11 شاشة مراقبة البيئة والاستدامة

5.12 شاشة التكاليف والربحية

شاشة صادرات المنتجات 5.13

شاشة ضمان الجودة والشهادات 5.14

شاشة البحث والتطوير 5.15

شاشة تقارير الإنتاج الشاملة 5.16

🍽️ المجموعة السادسة: شاشات إدارة التغذية (10 شاشات)

مركز التغذية الذكي 6.1

نظام التغذية المتطور:

```
┌──────────────────────────────────────────────────┐
│ 🌾 مركز إدارة التغذية الذكي                      │
│                                                  │
│ 📊 ملخص الاستهلاك اليومي:                        │
│ ┌──────────────┬──────────────────┐              │
│ │ 🌾 3,200 / 2,847 كجم │ علف أساسي │          │
│ │ 🌽 800 / 680 كجم │ ذرة صفراء │              │
│ │ 🥬 600 / 450 كجم │ برسيم حجازي │            │
│ │ 💊 120 / 89 كجم │ مكملات غذائية │           │
│ └──────────────┴──────────────────┘              │
│                                                  │
│ 🎯 برامج التغذية النشطة:                         │
│ ┌──────────────────────────────────────────┐     │
│ │ 🥛 برنامج الأبقار المنتجة (247 رأس)      │     │
│ │ ├ علف إنتاج: 8 كجم/رأس يومياً            │     │
│ │ ├ تبن قمح: 3 كجم/رأس يومياً              │     │
│ │ └ مكمل كالسيوم: 50 جم/رأس                │     │
│ │                                          │     │
│ │ 🤱 برنامج الحوامل والمرضعات (67 رأس)      │     │
│ │ ├ علف خاص: 12 كجم/رأس يومياً             │     │
│ │ ├ فيتامينات: 30 جم/رأس يومياً            │     │
│ │ └ معادن إضافية: 25 جم/رأس                │     │
│ │                                          │     │
│ │ 🐄 برنامج التسمين (189 رأس)              │     │
│ │ ├ علف تسمين: 15 كجم/رأس يومياً           │     │
│ │ ├ ذرة مكسرة: 4 كجم/رأس يومياً            │     │
│ │ └ بروتين: 200 جم/رأس يومياً              │     │
│ └──────────────────────────────────────────┘     │
│                                                  │
│ ⏰ جدولة الوجبات اليومية:                        │
│ 🌅 الفجر (05:30): وجبة أساسية                    │
│ 🌞 الضحى (10:00): مكملات ومياه                   │
│ 🌞 الظهر (13:00): وجبة رئيسية                    │
│ 🌇 المغرب (17:30): وجبة مسائية                   │
│ 🌙 العشاء (20:00): تبن ومكملات                   │
│                                                  │
│ 💧 نظام المياه الآلي:                            │
│ • إجمالي الاستهلاك: 15,600 لتر/يوم               │
│ • متوسط للرأس الواحد: 28 لتر/يوم                 │
│ • درجة حرارة المياه: 18-22°م                     │
│ • نقاط الشرب النشطة: 47 من 50 نقطة               │
│                                                  │
```

| 📈 تحليل كفاءة التحويل الغذائي: |
| ┌─────────────────────────────────────┐ |
	الأبقار الحلوب:	
	كل 1 كجم علف ← 2.8 لتر حليب	
	كفاءة: 94% (المعدل القياسي 85%)	
	العجول المسمنة:	
	كل 6.5 كجم علف ← 1 كجم وزن	
	كفاءة: 91% (المعدل القياسي 80%)	
└─────────────────────────────────────┘
🚨 تنبيهات التغذية:
🔴 نقص في مخزون البرسيم (يكفي 3 أيام)

تحليل نظام إدارة المزرعة المتكامل
وثيقة تحليلية احترافية للتطوير Full-Stack

🎯 ملخص تنفيذي

نظام إدارة المزرعة المتكامل هو منصة شاملة تهدف إلى رقمنة وأتمتة جميع عمليات المزرعة من إدارة الماشية والإنتاج إلى المبيعات والتوزيع، مع دعم العمل في البيئات النائية (Offline Mode) والربط مع أنظمة IoT وتقنيات RFID.

📋 الموديولات الرئيسية (Core Modules)

1️⃣ **موديول إدارة الماشية (Livestock Management)**

الشاشات الفرعية:
- **تسجيل الرؤوس الجديدة**
  - بيانات أساسية: رقم RFID، تاريخ الدخول، السلالة، الوزن، العمر
  - الحالة الصحية، مصدر الشراء، السعر
  - رفع صور، ربط بنظام GPS

- **إدارة التصنيفات والأنواع**
  - إنشاء أنواع جديدة من الماشية
  - تحديد جداول التطعيمات لكل نوع
  - برامج التغذية المخصصة
  - دورة الحمل والولادة (فترة الحمل، عدد المواليد المتوقع)

- **متابعة دورة الحياة**
  - مراحل النمو والتطور
  - سجل التزاوج والتلقيح

- سجل الولادات والنافقات
- متابعة الأوزان والقياسات

2️⃣ **موديول التكاثر والإنجاب (Breeding & Reproduction)**

الشاشات الفرعية:
- **جدولة التلقيح**
  - تحديد أوقات التلقيح المثلى
  - متابعة دورات الشبق
  - سجل الذكور المستخدمة في التلقيح

- **متابعة الحمل**
  - تسجيل تواريخ التلقيح
  - فحوصات الحمل الدورية
  - توقع مواعيد الولادة
  - تنبيهات ما قبل الولادة

- **إدارة الولادات**
  - تسجيل تفاصيل الولادة
  - حالة الأم والمولود
  - التدخلات البيطرية المطلوبة
  - إحصائيات معدلات النجاح

3️⃣ **موديول الرعاية البيطرية (Veterinary Care)**

الشاشات الفرعية:
- **جدول التطعيمات**
  - جدولة التطعيمات الدورية
  - متابعة التطعيمات المنجزة
  - تنبيهات التطعيمات المستحقة
  - سجل اللقاحات المستخدمة

- **السجل الصحي**
  - تسجيل الحالات المرضية
  - الفحوصات الدورية
  - العلاجات والأدوية المستخدمة
  - متابعة فترات النقاهة

- **إدارة الصيدلية البيطرية**
  - مخزون الأدوية واللقاحات
  - تواريخ الانتهاء والتنبيهات
  - استهلاك الأدوية لكل حالة
  - طلبات الشراء الآلية

4️⃣ **موديول الإنتاج (Production Management)**

الشاشات الفرعية:
- **إدارة الحليب**
  - ربط مع ماكينات الحلب الآلية
  - تسجيل كميات الحليب لكل رأس
  - جودة الحليب واختبارات الجودة
  - جدولة عمليات الحلب

- **إنتاج الصوف**
  - جدولة عمليات الجز
  - تسجيل كميات الصوف لكل رأس
  - تصنيف جودة الصوف
  - تخزين ومعالجة الصوف

- **إنتاج السماد الطبيعي**
  - تقدير كميات السماد المنتج
  - عمليات التجفيف والمعالجة
  - تعبئة وتخزين السماد
  - جودة السماد والتحاليل

5️⃣ **موديول التغذية (Feed Management)**

الشاشات الفرعية:
- **إدارة العلف**
  - أنواع العلف وتركيباتها
  - جدولة وجبات التغذية
  - حساب الكميات المطلوبة
  - متابعة استهلاك العلف

- **برامج التغذية المخصصة**
  - برامج تغذية حسب العمر والوزن
  - برامج تغذية للحوامل والمرضعات
  - برامج التسمين الخاصة
  - تحسين الأعلاف للإنتاجية

6️⃣ **موديول النقل والتوزيع (Transport & Distribution)**

الشاشات الفرعية:
- **أسطول المركبات**
  - بيانات المركبات وحالتها
  - جدولة الصيانة الدورية
  - تتبع المركبات بـ GPS
  - تكاليف التشغيل والوقود

- **إدارة الرحلات**
  - تخطيط رحلات النقل
  - تحميل وتفريغ المواشي
  - متابعة حالة المواشي أثناء النقل
  - تسجيل أوقات الوصول والمغادرة

- **نقل بين المزارع**
  - نقل من المزرعة الرئيسية للفرعية
  - تتبع ملكية المواشي بعد النقل
  - تسليم واستلام الشحنات
  - التوثيق والفواتير

7 **موديول المبيعات والمشتريات (Sales & Procurement)**

الشاشات الفرعية:
- **مبيعات الماشية**
  - قوائم أسعار حسب النوع والوزن
  - إدارة العملاء والموردين
  - عقود البيع والشراء
  - متابعة المدفوعات والمستحقات

- **مبيعات المنتجات**
  - بيع الحليب والصوف والسماد
  - عقود التوريد مع المزارع الفرعية
  - فواتير ومتابعة التحصيل
  - إحصائيات المبيعات

- **إعادة الشراء من المزارع الفرعية**
  - شراء المنتجات الثانوية (حليب، صوف، سماد)
  - متابعة جودة المنتجات المشتراة
  - حسابات التكلفة والربحية
  - عقود إعادة الشراء

8 **موديول المخازن والمستودعات (Inventory Management)**

الشاشات الفرعية:
- **مخزون العلف**
  - أرصدة العلف بأنواعه
  - تتبع تواريخ الانتهاء
  - تنبيهات إعادة الطلب
  - تكاليف التخزين

- **مخزون الأدوية**

- أرصدة الأدوية واللقاحات
- شروط التخزين والحفظ
- سجلات الاستخدام والصرف
- متابعة تواريخ الانتهاء

- **مخزون المنتجات**
- مخزون الحليب والصوف والسماد
- عمليات التعبئة والتغليف
- شروط التخزين والحفظ
- تتبع الدفعات والجودة

9️⃣ **موديول المالية والمحاسبة (Financial Management)**

الشاشات الفرعية:
- **الحسابات العامة**
- دفتر الأستاذ العام
- القيود المحاسبية
- الميزانية العمومية
- قائمة الدخل

- **إدارة التكاليف**
- تكاليف التغذية والرعاية
- تكاليف النقل والتشغيل
- حساب تكلفة الرأس الواحد
- تحليل الربحية

🔟 **موديول التقارير والتحليلات (Reports & Analytics)**

الشاشات الفرعية:
- **تقارير الإنتاج**
- تقارير إنتاج الحليب والصوف
- تحليل أداء القطيع
- مؤشرات الأداء الرئيسية
- مقارنات دورية

- **تقارير مالية**
- تقارير المبيعات والإيرادات
- تقارير التكاليف والمصروفات
- تحليل الربحية
- التدفقات النقدية

خريطة الترابط بين الموديولات

إدارة الماشية ←→ التكاثر والإنجاب ←→ الرعاية البيطرية
↓ ↓ ↓
إدارة التغذية ←→ موديول الإنتاج ←→ المخازن والمستودعات
↓ ↓ ↓
النقل والتوزيع ←→ المبيعات والمشتريات ←→ المالية والمحاسبة
↓ ↓ ↓
←← التقارير والتحليلات (مركز البيانات) ←←

التدفقات الأساسية:

1. **تدفق بيانات الماشية**: من التسجيل → الرعاية → الإنتاج → المبيعات
2. **تدفق المنتجات**: من الإنتاج → المخازن → المبيعات → التوزيع
3. **تدفق مالي**: من المشتريات/المبيعات → المحاسبة → التقارير
4. **تدفق معلومات**: جميع الموديولات → التقارير والتحليلات

🗄️ هيكل قواعد البيانات (Database Schema)

الجداول الأساسية (Core Tables):

1. **جدول الماشية (Livestock)**
```sql
CREATE TABLE livestock (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    rfid_tag VARCHAR(50) UNIQUE NOT NULL,
    livestock_type_id INT,
    gender ENUM('male', 'female'),
    birth_date DATE,
    entry_date DATE,
    weight DECIMAL(8,2),
    purchase_price DECIMAL(10,2),
    current_status ENUM('active', 'sold', 'deceased', 'quarantine'),
    farm_location_id INT,
    mother_id BIGINT,
    father_id BIGINT,
    created_at TIMESTAMP,
    updated_at TIMESTAMP
);
```

2. جدول أنواع الماشية (Livestock_Types)

```sql
```

```sql
CREATE TABLE livestock_types (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    breed VARCHAR(100),
    average_pregnancy_days INT,
    average_offspring_count INT,
    milking_capacity_per_day DECIMAL(6,2),
    wool_production_kg_per_year DECIMAL(6,2),
    feed_requirements TEXT,
    vaccination_schedule TEXT,
    created_at TIMESTAMP
);
```

3. جدول الحمل والولادة (Pregnancies)

```sql
CREATE TABLE pregnancies (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    female_livestock_id BIGINT,
    male_livestock_id BIGINT,
    breeding_date DATE,
    expected_birth_date DATE,
    actual_birth_date DATE,
    pregnancy_status ENUM('confirmed', 'suspected', 'completed', 'failed'),
    offspring_count INT,
    complications TEXT,
    veterinarian_notes TEXT,
    created_at TIMESTAMP
);
```

4. جدول الإنتاج (Production_Records)

```sql
```

```
CREATE TABLE production_records (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    livestock_id BIGINT,
    production_type ENUM('milk', 'wool', 'manure'),
    quantity DECIMAL(8,2),
    quality_grade VARCHAR(20),
    production_date DATE,
    milking_session_time TIME,
    notes TEXT,
    created_at TIMESTAMP
);
```

5. جدول الرعاية البيطرية (Veterinary_Records)

```
sql

CREATE TABLE veterinary_records (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    livestock_id BIGINT,
    record_type ENUM('vaccination', 'treatment', 'checkup'),
    medicine_id INT,
    dosage VARCHAR(50),
    veterinarian_name VARCHAR(100),
    treatment_date DATE,
    follow_up_date DATE,
    diagnosis TEXT,
    treatment_notes TEXT,
    cost DECIMAL(8,2),
    created_at TIMESTAMP
);
```

6. جدول النقل (Transport_Records)

```
sql
```

```
CREATE TABLE transport_records (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    vehicle_id INT,
    driver_id INT,
    departure_location_id INT,
    arrival_location_id INT,
    departure_time DATETIME,
    arrival_time DATETIME,
    livestock_count INT,
    transport_cost DECIMAL(8,2),
    gps_tracking_data TEXT,
    status ENUM('scheduled', 'in_transit', 'completed', 'cancelled'),
    created_at TIMESTAMP
);
```

7. جدول المبيعات (Sales_Records)

```
sql

CREATE TABLE sales_records (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    sale_type ENUM('livestock', 'milk', 'wool', 'manure'),
    item_id BIGINT,
    quantity DECIMAL(8,2),
    unit_price DECIMAL(8,2),
    total_amount DECIMAL(10,2),
    sale_date DATE,
    payment_status ENUM('pending', 'partial', 'completed'),
    contract_terms TEXT,
    created_at TIMESTAMP
);
```

🔌 واجهات API المطلوبة (50+ واجهة متخصصة)

APIs الأساسية:

1. Livestock Management API (15 واجهة)

```
GET    /api/livestock                  # قائمة الماشية مع فلترة متقدمة
POST   /api/livestock                  # إضافة رأس جديد
GET    /api/livestock/{id}             # تفاصيل رأس محدد
PUT    /api/livestock/{id}             # تحديث بيانات رأس
DELETE /api/livestock/{id}             # حذف/نقل رأس
POST   /api/livestock/bulk-import      # استيراد عدة رؤوس
PUT    /api/livestock/bulk-update      # تحديث مجموعة
GET    /api/livestock/search           # بحث متقدم
GET    /api/livestock/{id}/timeline    # الخط الزمني الكامل
GET    /api/livestock/{id}/health      # السجل الصحي الشامل
POST   /api/livestock/{id}/vaccination # تسجيل تطعيم
GET    /api/livestock/{id}/production  # سجل الإنتاج التفصيلي
GET    /api/livestock/{id}/genealogy   # شجرة النسب
POST   /api/livestock/{id}/transfer    # نقل الملكية
GET    /api/livestock/alerts           # تنبيهات الماشية
```

2. **Production Management API (12 واجهة)**

```
GET    /api/production/milk            # سجلات إنتاج الحليب
POST   /api/production/milk            # تسجيل حلبة جديدة
GET    /api/production/milk/{id}/quality # فحص جودة الحليب
POST   /api/production/milk/batch      # إنتاج دفعة
GET    /api/production/wool            # سجلات إنتاج الصوف
POST   /api/production/wool            # تسجيل جز جديد
GET    /api/production/wool/{id}/grading # تصنيف الصوف
GET    /api/production/manure          # سجلات السماد
POST   /api/production/manure          # تسجيل إنتاج سماد
GET    /api/production/analytics       # تحليلات الإنتاج
GET    /api/production/forecast        # توقعات الإنتاج
POST   /api/production/quality-test    # اختبارات الجودة
```

3. **Breeding & Reproduction API (10 واجهات)**

```
GET    /api/breeding/schedule              # جدولة التلقيح
POST   /api/breeding/mating                # تسجيل التزاوج
GET    /api/breeding/pregnancies           # سجل الحمل
PUT    /api/breeding/pregnancy/{id}        # تحديث حالة الحمل
POST   /api/breeding/birth                 # تسجيل ولادة
GET    /api/breeding/fertility-stats       # إحصائيات الخصوبة
GET    /api/breeding/genetic-analysis      # تحليل وراثي
POST   /api/breeding/artificial-insemination # تلقيح صناعي
GET    /api/breeding/breeding-values       # قيم التربية
GET    /api/breeding/estrus-detection      # كشف الشبق
```

4. **Veterinary Care API (15 واجهة)**

```
GET    /api/veterinary/records             # السجلات البيطرية
POST   /api/veterinary/examination         # فحص بيطري
GET    /api/veterinary/vaccinations        # جدول التطعيمات
POST   /api/veterinary/vaccination         # تسجيل تطعيم
GET    /api/veterinary/treatments          # العلاجات
POST   /api/veterinary/treatment           # تسجيل علاج
GET    /api/veterinary/diseases            # قاعدة الأمراض
POST   /api/veterinary/diagnosis           # تشخيص حالة
GET    /api/veterinary/medications         # الأدوية المتاحة
POST   /api/veterinary/prescription        # وصفة طبية
GET    /api/veterinary/health-alerts       # تنبيهات صحية
GET    /api/veterinary/quarantine          # إدارة العزل
POST   /api/veterinary/lab-test            # فحوصات مخبرية
GET    /api/veterinary/health-reports      # تقارير صحية
POST   /api/veterinary/emergency           # حالات طوارئ
```

5. **Feed Management API (8 واجهات)**

```
GET    /api/feed/inventory                 # مخزون العلف
POST   /api/feed/purchase                  # شراء علف
GET    /api/feed/nutrition-plans           # خطط التغذية
POST   /api/feed/feeding-schedule          # جدولة التغذية
GET    /api/feed/consumption               # استهلاك العلف
POST   /api/feed/recipe                    # تركيبة علف
GET    /api/feed/quality-analysis          # تحليل جودة العلف
GET    /api/feed/cost-analysis             # تحليل تكاليف
```

6. **Transport & Logistics API (12 واجهة)**

```
GET    /api/vehicles                    # أسطول المركبات
GET    /api/vehicles/{id}/location      # موقع المركبة الحالي
POST   /api/transport/schedule          # جدولة رحلة
PUT    /api/transport/{id}/status       # تحديث حالة الرحلة
GET    /api/transport/{id}/tracking     # تتبع الرحلة
POST   /api/transport/route-optimization  # تحسين المسارات
GET    /api/transport/driver-management   # إدارة السائقين
POST   /api/transport/loading           # تحميل البضائع
POST   /api/transport/delivery          # التسليم
GET    /api/transport/fuel-management   # إدارة الوقود
GET    /api/transport/maintenance       # صيانة المركبات
POST   /api/transport/emergency         # طوارئ النقل
```

7. IoT Integration API (واجهة 18)

```
POST   /api/iot/rfid/scan               # مسح RFID
GET    /api/iot/rfid/bulk-scan          # مسح مجموعة RFID
POST   /api/iot/rfid/register           # تسجيل RFID جديد
GET    /api/iot/sensors/temperature     # مستشعرات الحرارة
GET    /api/iot/sensors/humidity        # مستشعرات الرطوبة
GET    /api/iot/sensors/air-quality     # جودة الهواء
GET    /api/iot/milking/status          # حالة ماكينات الحلب
POST   /api/iot/milking/start           # بدء الحلب
POST   /api/iot/milking/stop            # إيقاف الحلب
POST   /api/iot/weight/reading          # قراءات الوزن
GET    /api/iot/scales/calibration      # معايرة الموازين
GET    /api/iot/gps/tracking            # تتبع GPS
POST   /api/iot/alerts/configure        # تكوين التنبيهات
GET    /api/iot/devices/status          # حالة الأجهزة
POST   /api/iot/devices/configure       # تكوين الأجهزة
GET    /api/iot/data/historical         # البيانات التاريخية
POST   /api/iot/calibration             # معايرة عامة
GET    /api/iot/connectivity            # حالة الاتصال
```

8. Sales & Procurement API (واجهة 14)

```
GET    /api/sales/livestock              # مبيعات الماشية
POST   /api/sales/livestock              # بيع ماشية
GET    /api/sales/products               # مبيعات المنتجات
POST   /api/sales/products               # بيع منتجات
GET    /api/sales/customers              # إدارة العملاء
POST   /api/sales/customer               # إضافة عميل
GET    /api/sales/contracts              # العقود
POST   /api/sales/contract               # إنشاء عقد
GET    /api/sales/invoices               # الفواتير
POST   /api/sales/invoice                # إنشاء فاتورة
GET    /api/sales/payments               # المدفوعات
POST   /api/sales/payment                # تسجيل دفع
GET    /api/procurement/suppliers        # الموردين
POST   /api/procurement/purchase-order   # أمر شراء
```

9. **Financial Management API (10 واجهات)**

```
GET    /api/finance/accounts             # الحسابات المالية
POST   /api/finance/transaction          # معاملة مالية
GET    /api/finance/budget               # الميزانية
POST   /api/finance/budget-allocation    # تخصيص ميزانية
GET    /api/finance/cost-analysis        # تحليل التكاليف
GET    /api/finance/profit-loss          # الأرباح والخسائر
GET    /api/finance/cash-flow            # التدفق النقدي
POST   /api/finance/expense              # تسجيل مصروف
GET    /api/finance/tax-calculations     # حسابات الضرائب
GET    /api/finance/roi-analysis         # تحليل العائد
```

10. **Inventory Management API (12 واجهة)**

```
GET    /api/inventory/items                عناصر المخزون #
POST   /api/inventory/item                 إضافة عنصر #
PUT    /api/inventory/item/{id}            تحديث عنصر #
GET    /api/inventory/stock-levels         مستويات المخزون #
POST   /api/inventory/stock-adjustment     تعديل المخزون #
GET    /api/inventory/movements            حركات المخزون #
POST   /api/inventory/transfer             نقل مخزون #
GET    /api/inventory/expiry-alerts        تنبيهات الانتهاء #
GET    /api/inventory/reorder-points       نقاط إعادة الطلب #
POST   /api/inventory/stocktaking          جرد المخزون #
GET    /api/inventory/valuation            تقييم المخزون #
POST   /api/inventory/batch-tracking       تتبع الدفعات #
```

11. **Farm Management API (واجهات 8)**

```
GET    /api/farms/locations                مواقع المزارع #
POST   /api/farms/location                 إضافة موقع #
GET    /api/farms/{id}/capacity            السعة الاستيعابية #
PUT    /api/farms/{id}/configuration       تكوين المزرعة #
GET    /api/farms/staff                    الموظفين #
POST   /api/farms/staff-assignment         تكليف موظف #
GET    /api/farms/facilities               المرافق #
POST   /api/farms/maintenance-schedule     جدولة الصيانة #
```

12. **Reporting & Analytics API (واجهة 15)**

```
GET    /api/reports/production             تقارير الإنتاج #
GET    /api/reports/financial              تقارير مالية #
GET    /api/reports/health                 تقارير صحية #
GET    /api/reports/breeding               تقارير التكاثر #
GET    /api/reports/feed-efficiency        كفاءة العلف #
GET    /api/analytics/kpi                  مؤشرات الأداء #
GET    /api/analytics/trends               التوجهات #
GET    /api/analytics/predictions          التنبؤات #
POST   /api/analytics/custom-query         استعلامات مخصصة #
GET    /api/analytics/benchmarking         المقارنات المعيارية #
GET    /api/analytics/seasonal-patterns    الأنماط الموسمية #
GET    /api/analytics/profitability        تحليل الربحية #
GET    /api/analytics/risk-assessment      تقييم المخاطر #
POST   /api/reports/export                 تصدير التقارير #
GET    /api/reports/scheduled              التقارير المجدولة #
```

13. **Offline Sync API** (8 واجهات)

```
POST   /api/sync/upload                  # رفع البيانات من الجهاز
GET    /api/sync/download                # تحميل التحديثات
GET    /api/sync/status                  # حالة المزامنة
POST   /api/sync/conflict-resolution     # حل التضارب
GET    /api/sync/pending-changes         # التغييرات المعلقة
POST   /api/sync/force-sync              # مزامنة إجبارية
GET    /api/sync/last-sync-time          # آخر وقت مزامنة
POST   /api/sync/batch-operations        # عمليات مجمعة
```

14. **User & Security API** (10 واجهات)

```
POST   /api/auth/login                   # تسجيل الدخول
POST   /api/auth/logout                  # تسجيل الخروج
POST   /api/auth/refresh-token           # تجديد الرمز
GET    /api/users/profile                # الملف الشخصي
PUT    /api/users/profile                # تحديث الملف
GET    /api/users/permissions            # الصلاحيات
POST   /api/users/role-assignment        # تعيين دور
GET    /api/audit/logs                   # سجلات التدقيق
POST   /api/security/report-incident     # بلاغ أمني
GET    /api/security/access-logs         # سجلات الوصول
```

15. **Notification & Alert API** (6 واجهات)

```
GET    /api/notifications                # قائمة الإشعارات
POST   /api/notifications/mark-read      # تمييز كمقروء
POST   /api/alerts/configure             # تكوين التنبيهات
GET    /api/alerts/active                # التنبيهات النشطة
POST   /api/alerts/acknowledge           # الإقرار بالتنبيه
GET    /api/alerts/history               # تاريخ التنبيهات
```

📱 تصميم الشاشات التفصيلي (80+ شاشة متخصصة)

🏠 المجموعة الأولى: شاشات **Dashboard** والتحكم الرئيسي (8 شاشات)

1.1 **Dashboard** الرئيسي المتقدم

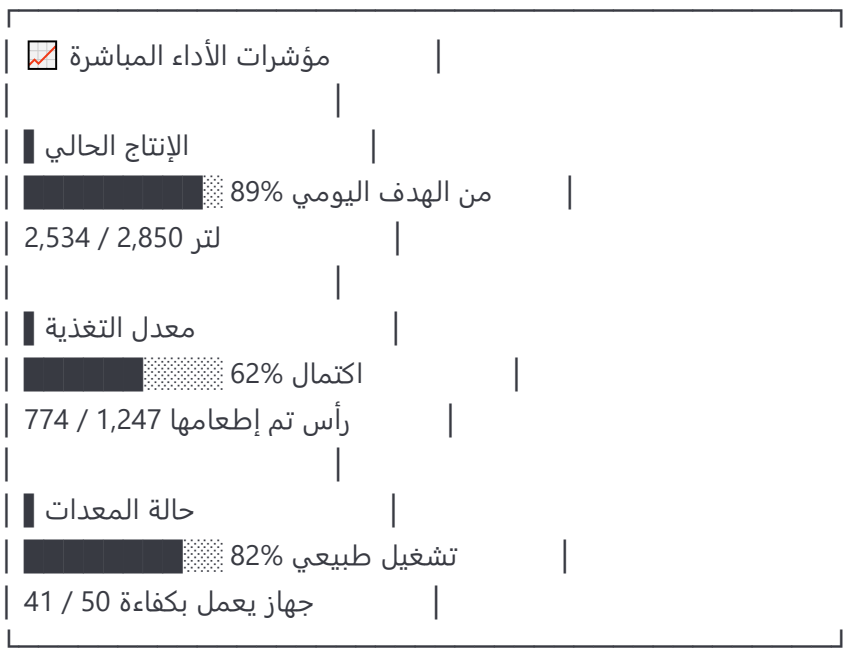
الحقول والعناصر:

```
┌─────────────────────────────────────────────┐
│ 📊 ملخص سريع (4 مربعات ملونة)               │
│ • إجمالي الماشية: 1,247 رأس                 │
│ • الإنتاج اليومي: 2,840 لتر                 │
│ • التنبيهات النشطة: 12 تنبيه                │
│ • الربح الشهري: 85,420 ريال                 │
├─────────────────────────────────────────────┤
│ 🗺️ خريطة المزرعة التفاعلية                  │
│ • مواقع القطعان مع ألوان مختلفة             │
│ • حالة المرافق (أخضر/أصفر/أحمر)             │
│ • مسارات المركبات المباشرة                  │
│ • نقاط الطعام والماء                        │
├─────────────────────────────────────────────┤
│ 🌡️ محطة الطقس المحلية                       │
│ • درجة الحرارة والرطوبة                     │
│ • سرعة الرياح واتجاهها                      │
│ • توقعات 3 أيام قادمة                       │
│ • تحذيرات الطقس الشديد                      │
├─────────────────────────────────────────────┤
│ 📅 جدول المهام اليومية                      │
│ • مواعيد التطعيمات (5 رؤوس)                 │
│ • جلسات الحلب (الساعة 6:00، 18:00)          │
│ • رحلات النقل المجدولة (2 رحلة)             │
│ • صيانة المعدات المستحقة                    │
└─────────────────────────────────────────────┘
```

المميزات التفاعلية:
- تحديث تلقائي كل 30 ثانية
- إشعارات منبثقة للطوارئ
- اختصارات سريعة للعمليات الشائعة
- رسوم بيانية متحركة للإنتاج

1.2 شاشة المؤشرات الحية (Live KPIs)

العناصر الأساسية:



الرسوم البيانية المباشرة:

- مخطط خطي لإنتاج الحليب (آخر 24 ساعة)
- مخطط دائري لتوزيع القطعان
- مؤشر حرارة لدرجات حرارة المرافق
- جدول زمني للأنشطة الجارية

1.3 مركز التحكم المركزي

أقسام الشاشة:

1.4 شاشة التقارير السريعة

1.5 لوحة المراقبة المالية

1.6 مركز الإشعارات المتقدم

1.7 شاشة الإعدادات العامة

1.8 لوحة الأمان والصلاحيات

المجموعة الثانية: شاشات إدارة الماشية (15 شاشة)

2.1 شاشة سجل الماشية الشامل

عناصر الشاشة الرئيسية:

خيارات العرض:
- عرض البطاقات (Cards View)
- عرض الجدول (Table View)
- عرض الخريطة (Map View)
- عرض الإحصائيات (Stats View)

2.2 شاشة إضافة رأس ماشية جديد

نموذج التسجيل المتقدم:

```
┌──────────────────────────────────────────────────┐
│ 📋 بيانات أساسية                                 │
│ RFID Tag: [_____] [📱 مسح]                       │
│ رقم الأذن: [_____] (اختياري)                     │
│ اسم/رقم الحيوان: [_____]                         │
│ نوع الماشية: [البقر ▼] [الغنم ▼] [الماعز ▼]      │
│ السلالة: [هولستين ▼] [جيرسي ▼] [أخرى ▼]          │
│ الجنس: ○ ذكر ○ أنثى                              │
│                                                  │
├──────────────────────────────────────────────────┤
│ 📅 معلومات العمر والتاريخ                        │
│ تاريخ الميلاد: [__/__/____] [📅]                 │
│ العمر التقديري: [_ سنة] [_ شهر]                  │
│ تاريخ دخول المزرعة: [__/__/____]                 │
│ مصدر الحصول: [شراء ▼] [مولود ▼] [هبة ▼]          │
│                                                  │
├──────────────────────────────────────────────────┤
│ ⚖️ القياسات الجسدية                              │
│ الوزن الحالي: [___] كجم [⚖️ وزن فوري]            │
│ الطول: [___] سم (اختياري)                        │
│ محيط الصدر: [___] سم (اختياري)                   │
│ حالة الجسم: [ضعيف] [متوسط] [جيد] [ممتاز]         │
│                                                  │
├──────────────────────────────────────────────────┤
│ 💰 المعلومات المالية                             │
│ سعر الشراء: [_____] ريال                         │
│ تكلفة النقل: [_____] ريال (اختياري)              │
│ رقم الفاتورة: [_____]                            │
│ اسم المورد: [_____]                              │
│                                                  │
├──────────────────────────────────────────────────┤
│ 🏥 الحالة الصحية الأولية                         │
│ حالة صحية: ○ سليم ○ مريض ○ حجر صحي               │
│ التطعيمات السابقة: [☑] جدري [☑] حمى قلاعية       │
│ ملاحظات طبية: [_____]                            │
│ الطبيب المعالج: [د. أحمد ▼]                      │
│                                                  │
├──────────────────────────────────────────────────┤
│ 👩‍👧 النسب والوراثة (للمواليد الجديدة)             │
│ الأم: [RF-Mother-001 ▼] [🔍 بحث]                 │
│ الأب: [RF-Father-005 ▼] [🔍 بحث]                 │
│ جيل: [الجيل الثالث ▼]                            │
```

أزرار التحكم:
[إلغاء] [مسح] [معاينة] [حفظ]

خصائص تقنية:
- التحقق من صحة البيانات فورياً
- حفظ تلقائي كل 30 ثانية
- دعم رفع صور متعددة
- ربط GPS تلقائي
- اقتراح أسماء ذكية

2.3 شاشة تفاصيل الحيوان الشاملة

تبويبات الشاشة الرئيسية:

2.4 شاشة البحث والفلترة المتقدمة

واجهة البحث المتطورة:

```
┌─────────────────────────────────────────────────────────┐
│ 🔍 البحث الذكي                                          │
│ [بحث عام: اكتب أي شيء...] [🎤 صوت]                     │
│ البحث بالرقم: [رقم الأذن/الاسم/RFID]                   │
│                                                         │
│ 🎛️ فلاتر متقدمة:                                       │
│ ├ نوع الحيوان: ☑ أبقار ☑ أغنام ☐ ماعز                  │
│ ├ الجنس: ☑ إناث ☑ ذكور                                 │
│ ├ العمر: من [_] إلى [_] سنة                            │
│ ├ الوزن: من [_] إلى [_] كجم                            │
│ ├ الحالة الصحية: ☑ سليم ☐ مريض ☐ حجر                   │
│ ├ حالة الإنتاج: ☑ منتج ☐ توقف ☐ جاف                    │
│ ├ المزرعة: [الكل ▼] [الرئيسية] [فرع أ]                 │
│ ├ تاريخ الدخول: من [__/__/__] إلى [__/__/__]           │
│ ├ مصدر الحصول: ☑ شراء ☑ مولود ☐ نقل                    │
│ └ السعر: من [____] إلى [____] ريال                     │
│                                                         │
│ 🔍 بحث متقدم:                                          │
│ • البحث الجغرافي (ضمن نطاق معين)                       │
│ • البحث بالباركود/QR Code                              │
│ • البحث بالصورة (التعرف البصري)                        │
│ • البحث بالصوت (أوامر صوتية)                           │
│                                                         │
│ 💾 فلاتر محفوظة:                                       │
│ [الأبقار المنتجة] [الحوامل] [للبيع]                    │
│ [+ حفظ فلتر جديد] [📁 إدارة الفلاتر]                   │
│                                                         │
│ 📊 نتائج البحث: 247 نتيجة                              │
│ [🔄 تحديث] [📤 تصدير] [📋 طباعة]                       │
└─────────────────────────────────────────────────────────┘
```

2.5 شاشة التتبع والحركة GPS

خريطة التتبع التفاعلية:

2.6 شاشة إدارة القطعان والمجموعات

2.7 شاشة النسب وشجرة العائلة

2.8 شاشة التقييم والتصنيف

2.9 شاشة نقل الملكية

2.10 شاشة الحجر الصحي

2.11 شاشة تصدير بيانات الماشية

2.12 شاشة الاستعلامات المتقدمة

2.13 شاشة التوثيق والصور

2.14 شاشة المقارنات والتحليل

2.15 شاشة التنبؤات الذكية

المجموعة الثالثة: شاشات التكاثر والإنجاب (12 شاشة) 🤱

3.1 شاشة إدارة دورة التكاثر

لوحة التحكم في التكاثر:

3.2 شاشة تسجيل التلقيح والتزاوج

نموذج تسجيل التلقيح:

┌──┐
│ 💞 تسجيل عملية التلقيح الجديدة │
│ │
│ 📋 بيانات الأنثى: │
│ الحيوان: [RF-234 - بقرة هولندية ▼] │
│ العمر: 3 سنة 8 شهر │
│ آخر ولادة: 15/08/2024 │
│ عدد الولادات السابقة: 2 │
│ حالة الدورة: جاهزة للتلقيح ✅ │
│ │
│ 📋 بيانات الذكر: │
│ نوع التلقيح: ○ طبيعي ● صناعي │
│ الطلق المستخدم: [رقم الطلقة: AI-789] │
│ مصدر الطلق: [مزرعة الأمل ▼] │
│ تاريخ إنتاج الطلق: 12/11/2024 │
│ جودة الطلق: ممتاز (حركة 85%) │
│ │
│ ⏰ توقيت التلقيح: │
│ التاريخ: [15/11/2024] [📅] │
│ الوقت: [08:30] صباحاً │
│ المدة: 15 دقيقة │
│ الطقس: مشمس، 24°م │
│ │
│ 👩‍⚕️ تفاصيل العملية: │
│ الطبيب البيطري: [د. محمد أحمد ▼] │
│ طريقة التلقيح: [داخل الرحم ▼] │
│ استخدام الهormون: نعم ☑️ │
│ نوع الهرمون: [GnRH] │
│ │
│ 📝 ملاحظات: │
│ [الحيوان كان هادئ ومتعاون أثناء العملية. │
│ تم استخدام مهدئ خفيف. توقع نتائج جيدة │
│ بناءً على حالة الحيوان الصحية الممتازة.] │
│ │
│ 📊 التوقعات: │
│ احتمالية الحمل: 78% (بناء على التاريخ) │
│ تاريخ الفحص المتوقع: 12/12/2024 │
│ تاريخ الولادة المتوقع: 22/08/2025 │
│ عدد المواليد المتوقع: 1 (احتمال 85%) │
│ │
│ 📅 المتابعة المطلوبة: │
│ ☑️ فحص الحمل بعد 21 يوم │

```
| ☑ فحص بالسونار بعد 45 يوم          |
| ☑ متابعة التغذية المخصصة           |
| ☑ جدولة التطعيمات اللازمة          |
|                                    |
| 💰 التكاليف:                        |
| تكلفة الطلق: 85 ريال               |
| أتعاب البيطري: 120 ريال            |
| الأدوية والهرمونات: 45 ريال        |
| إجمالي التكلفة: 250 ريال           |
└────────────────────────────────────┘

[طباعة التقرير 📋] [حفظ وجدولة المتابعة 💾]
```

3.3 شاشة متابعة الحمل والفحص

3.4 شاشة تسجيل الولادات

3.5 شاشة إدارة المواليد الجدد

3.6 شاشة التحليل الوراثي

3.7 شاشة جدولة التلقيح المتقدمة

3.8 شاشة إحصائيات الخصوبة

3.9 شاشة بنك الطلق المجمد

3.10 شاشة سجل النسل والأجيال

3.11 شاشة برنامج التحسين الوراثي

3.12 شاشة تقارير التكاثر المفصلة

🏥 المجموعة الرابعة: شاشات الرعاية البيطرية (18 شاشة)

4.1 لوحة الرعاية البيطرية المركزية

مركز التحكم البيطري:

```
┌──────────────────────────────────────────────────┐
│ 🏥 مركز الرعاية البيطرية                          │
│                                                  │
│ 🚨 حالات طوارئ نشطة:                              │
│ 🔴 (RF-123) حالة حرجة - صعوبة ولادة               │
│ 🟡 (RF-456) مراقبة - حمى خفيفة                    │
│ 🟠 (RF-789) متابعة - بعد العملية                  │
│                                                  │
│ 📊 إحصائيات اليوم:                                │
│ ┌──────────────┬──────────────────────────────┐  │
│ │ 🔬 23 فحص    | فحوصات اليوم                 ||
│ │ 💉 15 تطعيم  |  تطعيمات                    ||
│ │ 🏥 8 علاج    |  علاجات                      ||
│ │ 🏨 2 عملية   |  عمليات                      ||
│ └──────────────┴──────────────────────────────┘  │
│                                                  │
│ 📅 المواعيد المجدولة اليوم:                       │
│ 🕘 09:00 - فحص دوري (قطيع A)                      │
│ 🕙 10:30 - تطعيم ضد الجدري (15 رأس)               │
│ 🕐 13:00 - متابعة حالة مرضية (RF-234)             │
│ 🕒 15:00 - فحص الحمل بالسونار (8 رؤوس)           │
│                                                  │
│ 💊 حالة الصيدلية:                                 │
│ 🟢 مخزون جيد: 23 صنف                              │
│ 🟡 مخزون منخفض: 5 أصناف                           │
│ 🔴 انتهت الصلاحية: 2 أصناف                        │
│ 📦 طلبية واردة غداً: 12 صنف                       │
│                                                  │
│ 👥 الطاقم البيطري:                                │
│ 🟢 د. أحمد محمد - متاح                            │
│ 🟡 د. سارة علي - في عملية                         │
│ 🔴 د. محمود حسن - غير متاح                        │
│ 📞 طبيب الطوارئ: د. عمر (050-1234567)             │
└──────────────────────────────────────────────────┘
```

4.2 شاشة السجل الصحي الشامل

السجل الطبي المتكامل:

```
┌──────────────────────────────────────────────────────┐
│ 📋 السجل الصحي - RF-001 (بقرة هولندية)              │
│                                                      │
│ 🏷️ معلومات الحيوان:                                 │
│ الاسم: بقرة رقم 001 | العمر: 4 سنة                  │
│ الجنس: أنثى | الوزن: 485 كجم                        │
│ ✅ حالة صحية عامة: جيدة                             │
│                                                      │
│ 📊 مؤشرات حيوية حالية:                               │
│ ┌──────────────────┬──────────────────┐              │
│ │ درجة الحرارة     │ 38.5°م (طبيعي)   │              │
│ │ نبضات القلب      │ 70 نبضة/دقيقة   │              │
│ │ معدل التنفس      │ 22 نفس/دقيقة    │              │
│ │ ضغط الدم         │ طبيعي            │              │
│ └──────────────────┴──────────────────┘              │
│                                                      │
│ 📅 آخر الفحوصات والعلاجات:                          │
│ ┌──────────┬──────────────┬──────────┐               │
│ │ التاريخ  │ نوع العملية  │ النتيجة  │               │
│ ├──────────┼──────────────┼──────────┤               │
│ │15/11/2024│ ت            │          │               │
```

التصميم التقني:

الواجهة الأمامية (Frontend)
- **تقنية**: React.js / Vue.js مع PWA
- التوافق**: متجاوب لجميع الأجهزة**
- **العمل بلا اتصال**: Service Workers للعمل Offline
- الأمان**: تشفير البيانات والمصادقة المتعددة**

الواجهة الخلفية (Backend)
- **تقنية**: Node.js / Django / Laravel
- **قاعدة البيانات**: MySQL / PostgreSQL
- **الخوادم**: Cloud Infrastructure مع Load Balancing
- **APIs**: RESTful مع GraphQL للاستعلامات المعقدة

🗺️ خارطة طريق التنفيذ (Implementation Roadmap)

المرحلة الأولى (3-4 أشهر): الأسس والنواة
الأولوية القصوى:

- ✅ نظام المصادقة والأذونات
- ✅ إدارة الماشية الأساسية (CRUD)
- ✅ نظام RFID والتتبع
- ✅ واجهة المستخدم الأساسية
- ✅ قاعدة البيانات الأساسية

المرحلة الثانية (2-3 أشهر): الإنتاج والرعاية
التوسع الأساسي:
- ✅ موديول الإنتاج (حليب، صوف، سماد)
- ✅ النظام البيطري والصحي
- ✅ ربط ماكينات الحلب
- ✅ نظام التنبيهات والإشعارات
- ✅ التقارير الأساسية

المرحلة الثالثة (2-3 أشهر): النقل والتوزيع
التوسع اللوجستي:
- ✅ نظام النقل و GPS Tracking
- ✅ إدارة المركبات والأسطول
- ✅ ربط المزارع الرئيسية والفرعية
- ✅ نظام المبيعات والمشتريات
- ✅ العمل بلا اتصال (Offline Mode)

المرحلة الرابعة (2-3 أشهر): الذكاء والتكامل
التقنيات المتقدمة:
- ✅ تحليلات البيانات والذكاء الاصطناعي
- ✅ التنبؤ بالإنتاج والطلب
- ✅ تحسين العمليات تلقائياً
- ✅ تطبيق المحمول المتكامل
- ✅ APIs للأطراف الثالثة

المرحلة الخامسة (1-2 أشهر): التحسين والتطوير
التحسين والصيانة:
- ✅ اختبارات الأداء والأمان
- ✅ تحسين تجربة المستخدم
- ✅ التوثيق التقني الشامل
- ✅ التدريب ونقل المعرفة
- ✅ خطة الصيانة والتحديث

✅ معايير القبول (Acceptance Criteria)

المعايير الوظيفية:

1. **إدارة الماشية**
- ✅ إمكانية تسجيل رأس جديد في أقل من 2 دقيقة
- ✅ مسح RFID والتعرف على الرأس في أقل من 3 ثوانٍ
- ✅ تتبع حركة الماشية بدقة 95% أو أكثر
- ✅ حفظ جميع البيانات التاريخية بشكل دائم

2. **نظام الإنتاج**
- ✅ تسجيل كميات الحليب آلياً من ماكينات الحلب
- ✅ تتبع جودة المنتجات وفقاً للمعايير الدولية
- ✅ إنتاج تقارير إنتاج دقيقة بنسبة 99%
- ✅ تنبيهات فورية لأي انحراف في الإنتاج

3. **النقل والتتبع**
- ✅ تتبع مركبات GPS بتحديث كل دقيقة
- ✅ تسجيل رحلات النقل بالكامل
- ✅ تنبيهات الطوارئ في أقل من 30 ثانية
- ✅ توثيق عمليات التسليم والاستلام

4. **العمل بلا اتصال**
- ✅ العمل 72 ساعة دون اتصال إنترنت
- ✅ مزامنة البيانات خلال 5 دقائق من الاتصال
- ✅ حفظ جميع العمليات محلياً
- ✅ عدم فقدان أي بيانات أثناء انقطاع الاتصال

المعايير التقنية:

1. **الأداء**
- ✅ سرعة تحميل الصفحات أقل من 3 ثوانٍ
- ✅ استجابة النظام للطلبات أقل من 500ms
- ✅ إمكانية دعم 1000+ مستخدم متزامن
- ✅ تشغيل سلس على الأجهزة المحمولة

2. **الأمان**
- ✅ تشفير جميع البيانات الحساسة
- ✅ مصادقة ثنائية العامل
- ✅ سجلات تدقيق شاملة
- ✅ النسخ الاحتياطية التلقائية يومياً

3. **التوافق**
- ✅ دعم جميع المتصفحات الحديثة
- ✅ تطبيق محمول لأنظمة Android و iOS
- ✅ تكامل مع أنظمة ERP الموجودة
- ✅ APIs موثقة بالكامل للتكامل الخارجي

🚨 التحديات والمخاطر المحتملة

التحديات التقنية:

1. **الاتصالات في المناطق النائية**
- **التحدي**: ضعف أو انقطاع الإنترنت
- **الحل**: نظام العمل بلا اتصال مع مزامنة لاحقة
- **التنفيذ**: Service Workers و Local Database

2. **تكامل أجهزة IoT**
- **التحدي**: تنوع أجهزة الاستشعار والمعدات
- **الحل**: APIs موحدة ومحولات البروتوكولات
- **التنفيذ**: Middleware للتكامل مع أنظمة متعددة

3. **دقة بيانات RFID**
- **التحدي RFID**: فقدان أو تلف علامات
- **الحل**: نظام تتبع احتياطي ونسخ متعددة
- **التنفيذ**: QR Codes كبديل + نظام إنذار مبكر

المخاطر التجارية:

1. **مقاومة التغيير**
- **المخاطرة**: رفض المزارعين للتقنيات الجديدة
- **التخفيف**: تدريب مكثف وواجهة مستخدم بسيطة
- **الحل**: تنفيذ تدريجي مع دعم مستمر

2. **التكاليف التشغيلية**
- **المخاطرة**: ارتفاع تكاليف الاشتراك والصيانة
- **التخفيف**: نموذج تسعير مرن حسب حجم المزرعة
- **الحل**: عائد استثمار واضح ومقاس

🔧 متطلبات تقنية متخصصة

تكامل أجهزة IoT والاستشعار:

1. **أجهزة RFID المتقدمة**
```javascript
// مثال API للتكامل مع RFID Reader
const rfidIntegration = {
  async scanTag(readerId) {
```

```
    return {
      tagId: "EID123456789",
      timestamp: new Date(),
      signalStrength: 85,
      batteryLevel: 92,
      location: { lat: 24.7136, lng: 46.6753 }
    };
  },

  async bulkScan(readerId, duration = 30) {
    // مسح مجموعة من العلامات في وقت واحد
    return scannedTags;
  }
};
```

2. تكامل ماكينات الحلب الآلية

```
javascript

// تكامل مع أنظمة الحلب المختلفة (DeLaval, GEA, etc.)
const milkingSystemAPI = {
  async getMilkingData(cowId, sessionId) {
    return {
      cowId: "EID123456789",
      milkVolume: 28.5, // لتر
      milkingDuration: 420, // ثانية
      milkQuality: {
        fatContent: 3.8,
        proteinContent: 3.2,
        somaticCellCount: 150000,
        bacteria: "low"
      },
      timestamp: new Date(),
      milkingStallId: "stall_A_03"
    };
  }
};
```

3. أنظمة الوزن الذكية

```
javascript
```

```
const smartScaleAPI = {
  async getWeightReading(animalId) {
    return {
      animalId: "EID123456789",
      weight: 485.3, // كيلوجرام
      weightChange: +2.1, // تغيير عن آخر وزن
      confidence: 98, // دقة القراءة
      scalingConditions: "optimal",
      timestamp: new Date()
    };
  }
};
```

قواعد البيانات المتقدمة:

جداول إضافية مهمة:

8. جدول المزارع والفروع (Farms_Branches)

```
sql

CREATE TABLE farms_branches (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    type ENUM('main', 'branch'),
    parent_farm_id INT,
    manager_name VARCHAR(100),
    address TEXT,
    gps_coordinates POINT,
    capacity INT,
    current_livestock_count INT,
    establishment_date DATE,
    contact_info JSON,
    facilities JSON, -- معلومات المرافق المتاحة
    created_at TIMESTAMP
);
```

9. جدول تتبع الملكية (Ownership_Transfers)

```
sql
```

```
CREATE TABLE ownership_transfers (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    livestock_id BIGINT,
    from_farm_id INT,
    to_farm_id INT,
    transfer_date DATE,
    transfer_price DECIMAL(10,2),
    contract_terms TEXT,
    followup_agreement BOOLEAN,
    transfer_status ENUM('pending', 'completed', 'cancelled'),
    transport_record_id BIGINT,
    created_at TIMESTAMP
);
```

10. جدول المتابعة ما بعد البيع (Post_Sale_Monitoring)

```
sql

CREATE TABLE post_sale_monitoring (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    livestock_id BIGINT,
    current_owner_farm_id INT,
    monitoring_type ENUM('health', 'production', 'feeding'),
    monitoring_data JSON,
    monitoring_date DATE,
    recommendations TEXT,
    compliance_score INT, -- من 1-100
    created_at TIMESTAMP
);
```

📊 تقارير وتحليلات متقدمة

تقارير الإنتاجية الذكية:

1. تقرير كفاءة القطيع

```
sql
```

```sql
-- استعلام معقد لحساب كفاءة الإنتاج
WITH production_efficiency AS (
  SELECT
    l.id,
    l.rfid_tag,
    lt.name as breed,
    AVG(pr.quantity) as avg_daily_milk,
    COUNT(pr.id) as milking_sessions,
    (AVG(pr.quantity) / lt.milking_capacity_per_day * 100) as efficiency_percentage
  FROM livestock l
  JOIN livestock_types lt ON l.livestock_type_id = lt.id
  JOIN production_records pr ON l.id = pr.livestock_id
  WHERE pr.production_type = 'milk'
    AND pr.production_date >= DATE_SUB(NOW(), INTERVAL 30 DAY)
  GROUP BY l.id
)
SELECT * FROM production_efficiency
ORDER BY efficiency_percentage DESC;
```

2. تحليل التكاليف مقابل العوائد

```sql
sql

-- تحليل الربحية لكل رأس ماشية
WITH livestock_profitability AS (
  SELECT
    l.id,
    l.rfid_tag,
    -- التكاليف
    COALESCE(feed_costs.total, 0) as feeding_cost,
    COALESCE(vet_costs.total, 0) as veterinary_cost,
    -- العوائد
    COALESCE(milk_revenue.total, 0) as milk_income,
    COALESCE(wool_revenue.total, 0) as wool_income,
    -- الربح الصافي
    (COALESCE(milk_revenue.total, 0) + COALESCE(wool_revenue.total, 0) -
     COALESCE(feed_costs.total, 0) - COALESCE(vet_costs.total, 0)) as net_profit
  FROM livestock l
  -- ... استعلامات فرعية للتكاليف والعوائد
)
SELECT * FROM livestock_profitability
ORDER BY net_profit DESC;
```

📱 التطبيق المحمول المتخصص

مميزات التطبيق المحمول:

1. وضع العمل الحقلي (Field Mode)

- واجهة مبسطة للاستخدام بيد واحدة
- أزرار كبيرة وواضحة للعمل بالقفازات
- إدخال صوتي لتسجيل الملاحظات
- كاميرا سريعة للتوثيق

2. مسح RFID المدمج

```javascript
// تكامل مع NFC في الهواتف الذكية
const mobileRFID = {
  async startScanning() {
    if ('NDEFReader' in window) {
      const ndef = new NDEFReader();
      await ndef.scan();

      ndef.addEventListener("reading", ({ message, serialNumber }) => {
        // معالجة بيانات RFID
        this.processRFIDData(serialNumber, message);
      });
    }
  },

  processRFIDData(serialNumber, message) {
    // إرسال البيانات للخادم أو حفظها محلياً
    return {
      animalId: serialNumber,
      timestamp: new Date(),
      location: this.getCurrentLocation(),
      additionalData: message
    };
  }
};
```

3. الخرائط والملاحة المدمجة

```javascript
const farmNavigation = {
  async findAnimal(rfidTag) {
    const lastKnownLocation = await this.getLastKnownLocation(rfidTag);
    const currentLocation = await this.getCurrentLocation();

    return {
      distance: this.calculateDistance(currentLocation, lastKnownLocation),
      direction: this.calculateBearing(currentLocation, lastKnownLocation),
      estimatedWalkTime: this.estimateWalkTime(distance),
      mapRoute: this.generateRoute(currentLocation, lastKnownLocation)
    };
  }
};
```

🔐 الأمان وحماية البيانات

استراتيجية الأمان المتكاملة:

1. مصادقة متعددة المستويات

```javascript
```

```
const authenticationSystem = {
  // مصادقة بيومترية للوصول السريع
  async biometricAuth() {
    return await navigator.credentials.create({
      publicKey: {
        challenge: new Uint8Array(32),
        rp: { name: "Farm Management System" },
        user: { id: userId, name: userName, displayName: userDisplayName },
        pubKeyCredParams: [{ alg: -7, type: "public-key" }],
        authenticatorSelection: { authenticatorAttachment: "platform" }
      }
    });
  },

  // رموز OTP للعمليات الحساسة
  async generateOTP(operation) {
    return {
      code: this.generateSecureCode(),
      expiresIn: 300, // 5 دقائق
      operation: operation,
      timestamp: new Date()
    };
  }
};
```

2. تشفير البيانات الحساسة

```
javascript
```

```javascript
const dataEncryption = {
  async encryptSensitiveData(data, category) {
    const encryptionKey = await this.getEncryptionKey(category);
    const encryptedData = await crypto.subtle.encrypt(
      { name: "AES-GCM", iv: crypto.getRandomValues(new Uint8Array(12)) },
      encryptionKey,
      new TextEncoder().encode(JSON.stringify(data))
    );

    return {
      data: Array.from(new Uint8Array(encryptedData)),
      category: category,
      timestamp: new Date()
    };
  }
};
```

🌐 تكامل الأنظمة الخارجية

التكامل مع أنظمة ERP:

1. تصدير البيانات لأنظمة المحاسبة

```javascript
```

```javascript
const erpIntegration = {
  async exportToSAP(startDate, endDate) {
    const salesData = await this.getSalesData(startDate, endDate);
    const inventoryData = await this.getInventoryData(startDate, endDate);

    return {
      format: "SAP_IDOC",
      data: this.formatForSAP({
        sales: salesData,
        inventory: inventoryData,
        costCenters: await this.getCostCenters()
      }),
      exportDate: new Date()
    };
  },

  async syncWithQuickBooks() {
    // تزامن مع أنظمة QuickBooks
    const qbConnection = await this.establishQBConnection();
    await qbConnection.syncTransactions(this.getFinancialTransactions());
  }
};
```

2. تكامل مع أنظمة الطقس

```javascript
const weatherIntegration = {
  async getWeatherForecast(farmLocation) {
    const response = await fetch(`https://api.weather.com/v1/forecast?location=${farmLocation}`);
    const forecast = await response.json();

    // تحليل تأثير الطقس على الأنشطة الزراعية
    return {
      forecast: forecast,
      recommendations: this.generateWeatherRecommendations(forecast),
      alerts: this.checkWeatherAlerts(forecast)
    };
  }
};
```

📈 مؤشرات الأداء الرئيسية (KPIs)

مؤشرات الإنتاجية:

1. مؤشرات الإنتاج

- **متوسط إنتاج الحليب اليومي**: لتر/رأس/يوم
- **معدل كفاءة التحويل الغذائي**: كجم علف/لتر حليب
- **معدل الخصوبة**: نسبة الحمل الناجح
- **معدل البقاء على قيد الحياة**: نسبة النفوق

2. مؤشرات مالية

- **العائد على الاستثمار (ROI)**: نسبة مئوية سنوية
- **تكلفة الإنتاج لكل وحدة**: ريال/لتر أو ريال/كجم
- **هامش الربح الإجمالي**: نسبة مئوية
- **دورة رأس المال**: عدد الأيام

3. مؤشرات التشغيل

- **معدل استخدام الطاقة الاستيعابية**: نسبة الإشغال
- **كفاءة استخدام العلف**: معامل التحويل
- **معدل استخدام المعدات**: ساعات التشغيل
- **معدل الصيانة الوقائية**: نسبة الأعطال المتجنبة

خاتمة واستنتاجات 🎯

النتائج المتوقعة من تنفيذ النظام:

1. تحسينات تشغيلية

- **زيادة الإنتاجية**: 15-25% تحسن في الإنتاج
- **تقليل التكاليف**: 10-20% توفير في التكاليف التشغيلية
- **تحسين الجودة**: 30% تحسن في جودة المنتجات
- **كفاءة الوقت**: 40% توفير في وقت العمليات اليومية

2. فوائد استراتيجية

- **اتخاذ قرارات مبنية على البيانات**
- **تتبع شامل لسلسلة القيمة**
- **استباقية في حل المشاكل**

- **مرونة في التوسع والنمو**

3. عائد الاستثمار المتوقع

- **فترة الاسترداد**: 18-24 شهر
- **العائد السنوي**: 200-300% خلال 3 سنوات
- **توفير في العمالة**: 30-40% تقليل في ساعات العمل اليدوي
- **تحسن في دقة البيانات**: 95%+ دقة في التقارير

خطوات التنفيذ العملية 📞

المرحلة التحضيرية (شهر واحد):

1. تحليل المتطلبات التفصيلي

- جلسات عمل مع فريق المزرعة
- مسح المعدات الموجودة
- تقييم البنية التحتية
- وضع معايير القبول النهائية

2. إعداد البيئة التقنية

- إعداد الخوادم والبنية التحتية
- تثبيت قواعد البيانات
- إعداد أنظمة النسخ الاحتياطي
- اختبار الشبكات والاتصالات

3. تشكيل فريق العمل

- **مطور Backend**: خبير Node.js/Django
- **مطور Frontend**: متخصص React/Vue.js
- **مطور Mobile**: Flutter/React Native
- **مختص IoT**: تكامل الأجهزة والاستشعار
- **محلل قواعد بيانات**: تحسين الأداء
- **مختص UX/UI**: تصميم التجربة
- **مدير مشروع**: متابعة التنفيذ

هذه الوثيقة تشكل الأساس الشامل لتطوير نظام إدارة المزرعة المتكامل. يمكن للفرق التقنية البدء في التنفيذ بناء على هذا التحليل المفصل.

آخر تحديث: ${new Date().toLocaleDateString('ar-SA')}
المحلل: System Analyst - Farm Management Expert
حالة الوثيقة: جاهزة للتنفيذ