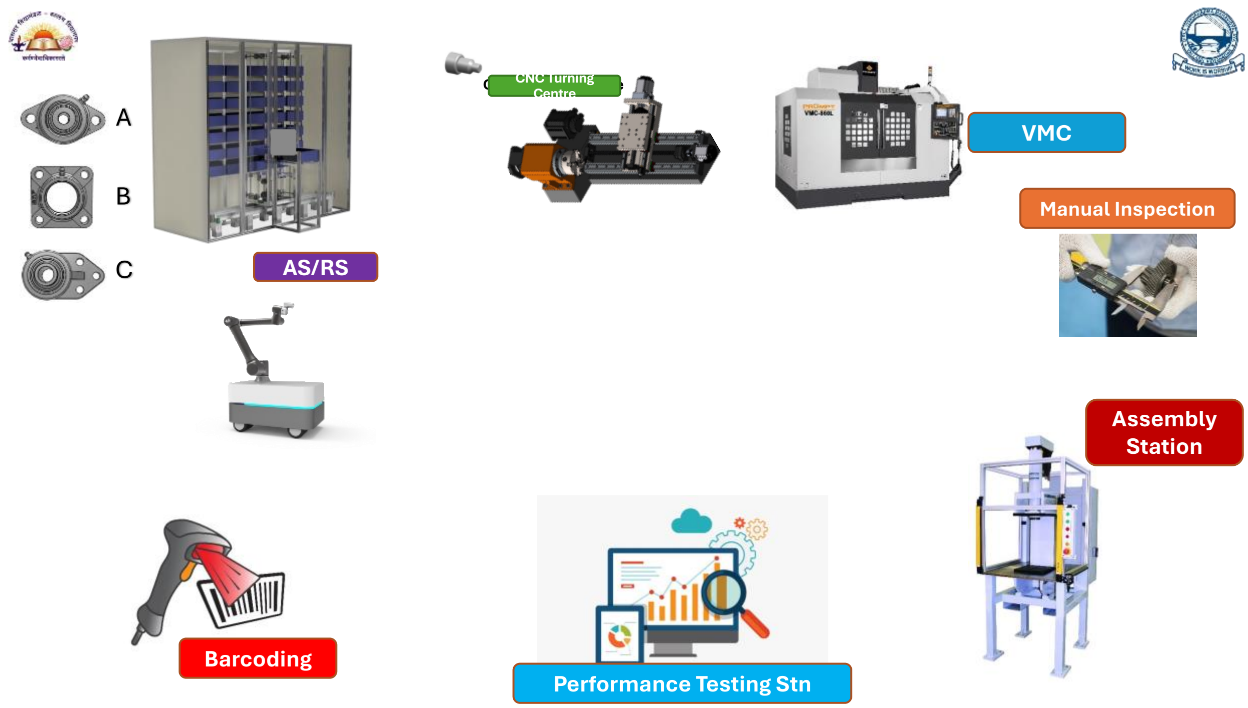
Birla Vishvakarma Mahavidyala

Center of Excellence in Digital Manufacturing



Summer Internship Program – June 2025

List of Projects

Project: 2025.06.1

Title of the Project:

"Retrofitting and IIoT Integration of TRIAC-PC CNC Machining Center"

Objective:

- To complete the retrofitting of the TRIAC-PC CNC Machining Center by integrating electronic and mechanical components, including controllers, motors, sensors, and control panel.
- To establish IIoT connectivity using protocols such as OPC-UA and Ethernet/IP for remote monitoring, data acquisition, and performance analytics.
- To test and validate the CNC operations and IIoT system for reliable machining and real-time data visualization.

Present Status:

- **Mechanical Components:** Design and manufacturing of mechanical components required for retrofitting is completed.
- **Electronic Components:** Procurement of controllers, motors, sensors, and related hardware is in progress.
- **Control Panel:** Design is finalized; fabrication and wiring are yet to begin.
- **Software Programming:** Yet to be initiated (PLC/CNC and IIoT interfaces).

Technologies to be Used:

Hardware:

- **CNC Controllers:** Compatible with TRIAC-PC spindle and axis control.
- **Motors:** Servo motors for X, Y, Z axes and spindle control.
- **Sensors:** Limit switches, encoders, safety interlocks.
- **IIoT Modules:** Devices supporting OPC-UA or Ethernet/IP communication.
- **Control Panel Components:** Power supplies, circuit protection, terminal blocks, relays.

Software:

- **CNC Programming:** G-code and motion sequencing for machining operations.

- **Control System Design:** Ladder logic or structured text (as per controller type).
- **IIoT Middleware:** KEPServerEX / Node-RED / ThingWorx for protocol handling.
- **Monitoring Dashboard:** Real-time analytics using Power BI / Grafana / ThingWorx.
- **Panel Design:** EPLAN / AutoCAD Electrical for wiring and layout design.

Key Deliverables:

1. **Completed Retrofitting of TRIAC-PC:**
    - Integration of all mechanical and electronic components.
    - Fully assembled and functional control panel.
2. **CNC Operation Enablement:**
    - Verified motion control of axes and spindle through the new controller setup.
    - Basic job execution using CNC programming.
3. **IIoT Connectivity:**
    - OPC-UA/Ethernet-IP enabled data transmission from machine to dashboard.
    - Acquisition of key machine parameters like cycle time, axis position, spindle load, etc.
4. **Monitoring & Analytics Dashboard:**
    - Real-time display of machine status and performance metrics.
    - Historical trend analysis and fault/event logging.
5. **Testing and Validation Report:**
    - Functional testing of machining operations.
    - Final documentation covering electrical drawings, IIoT configuration, and optimization recommendations.

Project: 2025.06.2

Title of the Project:

"Design and Implementation of Inventory Management and Transaction Monitoring System for AS/RS"

Objective:

- To develop a system that logs and maintains real-time inventory records of items stored in and retrieved from the Automatic Storage and Retrieval System (AS/RS).
- To integrate the AS/RS PLC control logic with a user interface for inventory input, monitoring, and reporting.
- To implement transaction tracking with unique identification of each storage/retrieval operation.
- To enable generation of inventory reports, alert messages for errors, and manual override functions.

Present Status:

- **Mechanical Assembly:** 90% of the AS/RS mechanical setup is complete.
- **Electronics:** PLC, motors, and sensors are procured and control panel wiring is completed.
- **Control Logic:** Basic PLC programming for storage and retrieval operations is functional.
- **System Integration:** Core motion and sensor functionality are tested and verified.

Technologies to be Used:

- **PLC Programming:** Siemens S7-1200 / Allen-Bradley / Omron (depending on existing setup) using Ladder Logic or Structured Text.
- **SCADA/HMI:** WinCC / FactoryTalk / NB Designer / Custom Python GUI using `tkinter` or `PyQt`.
- **Database:** SQLite / MySQL for storing transaction data (item ID, timestamp, status, bin location).
- **Communication:** OPC-UA / Modbus TCP for PLC–PC/SCADA communication.
- **Optional Web Interface:** Node-RED / Flask dashboard for remote monitoring (if required).
- **Barcode/RFID (Optional):** For automated item identification in inventory records.

Key Deliverables:

1. **Digital Inventory Database** with real-time updates on item storage and retrieval.
2. **User Interface (SCADA/HMI/Python GUI)** for visualizing current storage status and managing operations.
3. **PLC Program Enhancement** to generate and send transaction events to the database.
4. **Transaction Log System** with unique IDs, timestamps, storage bin positions, and action types (store/retrieve).
5. **Error Handling and Notification System** (e.g., sensor faults, bin occupancy conflicts).
6. **Inventory Report Generator** (daily/weekly/monthly reports exportable in CSV/PDF).
7. **Optional Integration of Barcode/RFID System** for automation of item identification.
8. **Demonstration of Full Cycle Operation** including automatic logging and retrieval using GUI.

Project: 2025.06.3

Title of the Project:

"ROS2-Based Autonomous Navigation System Development for an Indoor Autonomous Mobile Robot (AMR)"

Objective:

- To complete the integration of electronic circuits and sensor systems (LiDAR and IMU) on the manufactured AMR platform.
- To develop and implement autonomous navigation using **ROS2**, including mapping, localization, path planning, and obstacle avoidance.
- To validate the AMR's performance through real-time testing in a confined indoor environment.

Present Status:

- **CAD Design:** Completed.
- **Mechanical Manufacturing and Assembly:** Fully completed.
- **Electronics Integration:** Sensors and control circuitry yet to be installed and wired.
- **Software Development:** ROS2 programming, sensor drivers, and navigation stack setup pending.

Technologies to be Used:

Hardware:

- **Controller:** NVIDIA Jetson Orin Nano / AGX.
- **Sensors:**
  - **LiDAR** (e.g., RPLidar A2): For mapping and obstacle detection.
  - **IMU** (e.g., BNO055, MPU-9250): For orientation and localization support.
- **Motor Drivers and Power System:** For robot actuation and regulated power supply.
- **Connectivity:** UART/USB/Serial interfaces for sensor-controller communication.

Software & Frameworks:

- **OS:** Ubuntu 22.04 LTS.
- **Middleware:** ROS2 (e.g., Humble Hawksbill or Iron Irwini).
- **SLAM & Mapping:** `slam_toolbox` or `Cartographer`.

- **Navigation Stack:** `nav2` (includes AMCL, global and local planners, obstacle avoidance).
- **Sensor Drivers:** ROS2-compatible packages for LiDAR and IMU.
- **Visualization:** RViz2 for real-time visualization of robot environment and diagnostics.

Key Deliverables:

1. **Electronics Integration:**
    - Mounting and wiring of LiDAR, IMU, motor drivers, and power circuits.
    - Verification of sensor output and data communication with Jetson Orin.
2. **ROS2-Based Navigation System:**
    - ROS2 workspace with nodes for sensor drivers, SLAM, localization, and path planning.
    - Launch files and parameter tuning for reliable indoor navigation.
3. **Autonomous Navigation Features:**
    - Dynamic obstacle detection and avoidance.
    - Point-to-point movement with map-based localization.
    - Testing in confined indoor space for repeatable performance.
4. **Performance Testing and Validation:**
    - Test reports showing accuracy, repeatability, and robustness of navigation.
    - Comparison of real-time path execution with planned trajectories.
5. **Final Documentation:**
    - System architecture, ROS2 package hierarchy, wiring schematics.
    - Setup and operation manual for future users.
    - Recommendations for future improvements (e.g., multi-floor navigation, camera fusion).

Project: 2025.06.4

Title of the Project:

"Vision-Based Pick and Place Automation Using OMRON RT6 Collaborative Robot (COBOT)"

Objective:

1. To develop a program for the **OMRON RT6 COBOT** to perform pick-and-place tasks for various types of objects.
2. To integrate and configure the vision system for object detection, classification, and precise localization.
3. To optimize the pick-and-place motion for efficiency and accuracy, ensuring seamless object handling.
4. To test and validate the system in a real-time environment for diverse object sizes, shapes, and orientations.

Present Status:

1. **Hardware**:
    - OMRON RT6 COBOT (RT6 5004BRT6-0107001) has been purchased and is available.
    - Vision system integrated with the COBOT is available for use.
2. **Software and Programming**:
    - Programming and configuration for pick-and-place tasks are yet to be developed.
    - Vision-based object recognition and positioning algorithms need to be implemented.

Technologies to be Used:

1. **Hardware**:
    - OMRON RT6 COBOT with integrated vision system.
    - Grippers for object handling (as per object type).
2. **Software**:
    - **OMRON TMFlow or Sysmac Studio**: For programming and controlling the COBOT.
    - Vision system software: For object detection, classification, and localization.
3. **Algorithms and Methods**:

- Image processing techniques (e.g., edge detection, blob detection) for object recognition.
- Path planning for pick-and-place operations.
- Optimization algorithms for efficient object handling.

Key Deliverables:

1. **Fully Programmed COBOT**: Capable of performing pick-and-place operations for a variety of objects.
2. **Vision System Integration**: Object detection and precise placement using the integrated vision system.
3. **Performance Testing**: Evaluation under different scenarios for accuracy, efficiency, and reliability.

Project: 2025.06.5

Title of the Project**:**

Title:
"Development of an Interactive Virtual Tour for the Center of Excellence in Digital Manufacturing"

Objective:

- To design and develop an immersive, interactive **Virtual Reality (VR) Tour** showcasing the complete workflow and infrastructure of the Center of Excellence in Digital Manufacturing.
- To digitally model and animate each station (MIRAC-PC, TRIAC-PC, Assembly, Manual Inspection, Performance Measurement, AS/RS) and the movement of the **AMR with mounted COBOT**.
- To integrate VR navigation, hotspots, and information overlays to provide users with an educational and engaging experience.
- To enable accessibility of the tour through VR headset and optionally via web browser or desktop mode.

Present Status:

- All physical stations (MIRAC-PC, TRIAC-PC, Assembly Station, etc.) are in various stages of setup or completion.
- CAD models of machines and stations are available or under preparation.
- AMR platform and COBOT integration are in progress.
- VR headset is available and ready for deployment.
- No virtual environment or software development initiated yet.

Technologies to be Used:

Hardware:

- **VR Headset:** Meta Quest / HTC Vive / Any available headset for immersive deployment.
- **Workstation:** PC or laptop with VR development and rendering capability.

Software:

- **3D Modeling Tools:** Blender / SolidWorks / Fusion 360 (for CAD to 3D conversion).
- **Game Engine / VR Platform:** Unity 3D or Unreal Engine (for VR environment development).
- **Programming Language:** C# (for Unity) or Blueprint/optional C++ (for Unreal Engine).
- **VR Integration SDKs:** SteamVR, Oculus SDK (depending on headset).
- **Optional Web Deployment:** WebXR / Mozilla Hubs (for browser-based access).

Key Deliverables:

1. **Digital 3D Models of Stations:**
   - Import or recreate MIRAC-PC, TRIAC-PC, AMR with COBOT, AS/RS, and other stations in Unity/Unreal.
   - Optimize models for real-time rendering in VR.
2. **Virtual Environment Creation:**
   - Construct the complete layout of the Center of Excellence with walk-through capability.
   - Include textures, lighting, and interaction-ready elements.
3. **Interactive Elements:**
   - Hotspots with technical information, animations (e.g., AMR moving, COBOT picking parts).
   - Pop-up media: diagrams, images, videos explaining station functions.
4. **VR Integration:**
   - Compatible experience for VR headset (6DOF navigation).
   - Smooth controls, user-friendly UI, and performance optimization.
5. **Optional Desktop/Web Version:**
   - Lightweight version for non-VR users via keyboard/mouse or browser.
6. **Testing and Deployment:**
   - Functional testing on VR hardware for immersion, navigation, and accuracy.
   - Collection of feedback from faculty and student users.
7. **Documentation:**
   - Developer guide with source files and build instructions.
   - User manual for operating the VR experience.
   - Final report on implementation steps and future enhancement scope (e.g., adding live data from stations).

Project: 2025.06.6

Title of the Project:

"Augmented Reality (AR) Application Development for MIRAC-PC at the Center of Excellence in Digital Manufacturing"

Objective:

- To develop a mobile-based Augmented Reality (AR) application that enhances user engagement by displaying interactive information about the **MIRAC-PC** when a QR code at the station is scanned.
- To integrate 3D visualizations, animations, technical descriptions, and videos to help students, faculty, and visitors understand the function and operation of the station.
- To ensure smooth deployment on Android/iOS devices and compatibility with commonly available AR headsets or smart devices.

Present Status:

- The **[Station Name]** has been designed or is in the final phase of setup under the Center of Excellence in Digital Manufacturing.
- Physical QR code space can be allocated at the station.
- No AR content or software development has been initiated yet.

Technologies to be Used:

Hardware:

- Smartphones/Tablets (Android/iOS).
- Optional AR headset (e.g., Hololens, Magic Leap – if available).

Software:

- **AR Development Platform:** Unity 3D with Vuforia Engine or 8thWall (WebAR) / ZapWorks.
- **3D Modeling Tools:** Blender / SolidWorks / Fusion 360 (to export CAD as .fbx/.glb for Unity).
- **Programming Language:** C# (for Unity) or JavaScript (for WebAR).
- **QR Code Tools:** Embedded QR codes for station-linked AR triggers.
- **Optional CMS/Database:** Firebase or Google Sheets for dynamically updating content.

Key Deliverables:

1. **QR Code Integration:**
   - Unique QR code for **[Station Name]** that launches AR experience on device.
2. **3D Visualization of Station:**
   - CAD-to-AR conversion of the machine or setup with visual fidelity.
   - Annotated 3D model showing parts, functionality, and interactions.
3. **Interactive AR Content:**
   - Text, audio, or video descriptions.
   - Process animations or guided step-by-step sequences (e.g., material flow, inspection method).
4. **Cross-Platform AR Application:**
   - APK or WebAR link for Android and iOS deployment.
   - Simple, intuitive user interface for scan-trigger-view experience.
5. **Testing & Feedback:**
   - Usability testing with faculty/students.
   - Performance tuning for various mobile devices.
6. **Final Documentation:**
   - User manual for app operation.
   - Developer documentation including source code, assets, and build instructions.
   - Suggestions for future scalability (e.g., multilingual support, integration with live sensor data).

Project: 2025.06.7

Title of the Project:

"Augmented Reality (AR) Application Development for TRIAC-PC at the Center of Excellence in Digital Manufacturing"

Objective:

- To develop a mobile-based Augmented Reality (AR) application that enhances user engagement by displaying interactive information about the **TRIAC-PC** when a QR code at the station is scanned.
- To integrate 3D visualizations, animations, technical descriptions, and videos to help students, faculty, and visitors understand the function and operation of the station.
- To ensure smooth deployment on Android/iOS devices and compatibility with commonly available AR headsets or smart devices.

Present Status:

- The **[Station Name]** has been designed or is in the final phase of setup under the Center of Excellence in Digital Manufacturing.
- Physical QR code space can be allocated at the station.
- No AR content or software development has been initiated yet.

Technologies to be Used:

Hardware:

- Smartphones/Tablets (Android/iOS).
- Optional AR headset (e.g., Hololens, Magic Leap – if available).

Software:

- **AR Development Platform:** Unity 3D with Vuforia Engine or 8thWall (WebAR) / ZapWorks.
- **3D Modeling Tools:** Blender / SolidWorks / Fusion 360 (to export CAD as .fbx/.glb for Unity).
- **Programming Language:** C# (for Unity) or JavaScript (for WebAR).
- **QR Code Tools:** Embedded QR codes for station-linked AR triggers.
- **Optional CMS/Database:** Firebase or Google Sheets for dynamically updating content.

Key Deliverables:

1. **QR Code Integration:**
   - Unique QR code for **[Station Name]** that launches AR experience on device.
2. **3D Visualization of Station:**
   - CAD-to-AR conversion of the machine or setup with visual fidelity.
   - Annotated 3D model showing parts, functionality, and interactions.
3. **Interactive AR Content:**
   - Text, audio, or video descriptions.
   - Process animations or guided step-by-step sequences (e.g., material flow, inspection method).
4. **Cross-Platform AR Application:**
   - APK or WebAR link for Android and iOS deployment.
   - Simple, intuitive user interface for scan-trigger-view experience.
5. **Testing & Feedback:**
   - Usability testing with faculty/students.
   - Performance tuning for various mobile devices.
6. **Final Documentation:**
   - User manual for app operation.
   - Developer documentation including source code, assets, and build instructions.
   - Suggestions for future scalability (e.g., multilingual support, integration with live sensor data).

Project: 2025.06.8

Title of the Project:

"Augmented Reality (AR) Application Development for Assembly Station at the Center of Excellence in Digital Manufacturing"

Objective:

- To develop a mobile-based Augmented Reality (AR) application that enhances user engagement by displaying interactive information about the **Assembly Station** when a QR code at the station is scanned.
- To integrate 3D visualizations, animations, technical descriptions, and videos to help students, faculty, and visitors understand the function and operation of the station.
- To ensure smooth deployment on Android/iOS devices and compatibility with commonly available AR headsets or smart devices.

Present Status:

- The **[Station Name]** has been designed or is in the final phase of setup under the Center of Excellence in Digital Manufacturing.
- Physical QR code space can be allocated at the station.
- No AR content or software development has been initiated yet.

Technologies to be Used:

Hardware:

- Smartphones/Tablets (Android/iOS).
- Optional AR headset (e.g., Hololens, Magic Leap – if available).

Software:

- **AR Development Platform:** Unity 3D with Vuforia Engine or 8thWall (WebAR) / ZapWorks.
- **3D Modeling Tools:** Blender / SolidWorks / Fusion 360 (to export CAD as .fbx/.glb for Unity).
- **Programming Language:** C# (for Unity) or JavaScript (for WebAR).
- **QR Code Tools:** Embedded QR codes for station-linked AR triggers.
- **Optional CMS/Database:** Firebase or Google Sheets for dynamically updating content.

Key Deliverables:

1. **QR Code Integration:**
    - Unique QR code for **[Station Name]** that launches AR experience on device.
2. **3D Visualization of Station:**
    - CAD-to-AR conversion of the machine or setup with visual fidelity.
    - Annotated 3D model showing parts, functionality, and interactions.
3. **Interactive AR Content:**
    - Text, audio, or video descriptions.
    - Process animations or guided step-by-step sequences (e.g., material flow, inspection method).
4. **Cross-Platform AR Application:**
    - APK or WebAR link for Android and iOS deployment.
    - Simple, intuitive user interface for scan-trigger-view experience.
5. **Testing & Feedback:**
    - Usability testing with faculty/students.
    - Performance tuning for various mobile devices.
6. **Final Documentation:**
    - User manual for app operation.
    - Developer documentation including source code, assets, and build instructions.
    - Suggestions for future scalability (e.g., multilingual support, integration with live sensor data).

Project: 2025.06.9

Title of the Project:

"Augmented Reality (AR) Application Development for AS/RS at the Center of Excellence in Digital Manufacturing"

Objective:

- To develop a mobile-based Augmented Reality (AR) application that enhances user engagement by displaying interactive information about the **AS/RS** when a QR code at the station is scanned.
- To integrate 3D visualizations, animations, technical descriptions, and videos to help students, faculty, and visitors understand the function and operation of the station.
- To ensure smooth deployment on Android/iOS devices and compatibility with commonly available AR headsets or smart devices.

Present Status:

- The **[Station Name]** has been designed or is in the final phase of setup under the Center of Excellence in Digital Manufacturing.
- Physical QR code space can be allocated at the station.
- No AR content or software development has been initiated yet.

Technologies to be Used:

Hardware:

- Smartphones/Tablets (Android/iOS).
- Optional AR headset (e.g., Hololens, Magic Leap – if available).

Software:

- **AR Development Platform:** Unity 3D with Vuforia Engine or 8thWall (WebAR) / ZapWorks.
- **3D Modeling Tools:** Blender / SolidWorks / Fusion 360 (to export CAD as .fbx/.glb for Unity).
- **Programming Language:** C# (for Unity) or JavaScript (for WebAR).
- **QR Code Tools:** Embedded QR codes for station-linked AR triggers.
- **Optional CMS/Database:** Firebase or Google Sheets for dynamically updating content.

Key Deliverables:

1. **QR Code Integration:**
    - Unique QR code for **[Station Name]** that launches AR experience on device.
2. **3D Visualization of Station:**
    - CAD-to-AR conversion of the machine or setup with visual fidelity.
    - Annotated 3D model showing parts, functionality, and interactions.
3. **Interactive AR Content:**
    - Text, audio, or video descriptions.
    - Process animations or guided step-by-step sequences (e.g., material flow, inspection method).
4. **Cross-Platform AR Application:**
    - APK or WebAR link for Android and iOS deployment.
    - Simple, intuitive user interface for scan-trigger-view experience.
5. **Testing & Feedback:**
    - Usability testing with faculty/students.
    - Performance tuning for various mobile devices.
6. **Final Documentation:**
    - User manual for app operation.
    - Developer documentation including source code, assets, and build instructions.
    - Suggestions for future scalability (e.g., multilingual support, integration with live sensor data).

Project: 2025.06.10

Title of the Project:

"Augmented Reality (AR) Application Development for AMR at the Center of Excellence in Digital Manufacturing"

Objective:

- To develop a mobile-based Augmented Reality (AR) application that enhances user engagement by displaying interactive information about the **AMR** when a QR code at the station is scanned.
- To integrate 3D visualizations, animations, technical descriptions, and videos to help students, faculty, and visitors understand the function and operation of the station.
- To ensure smooth deployment on Android/iOS devices and compatibility with commonly available AR headsets or smart devices.

Present Status:

- The **[Station Name]** has been designed or is in the final phase of setup under the Center of Excellence in Digital Manufacturing.
- Physical QR code space can be allocated at the station.
- No AR content or software development has been initiated yet.

Technologies to be Used:

Hardware:

- Smartphones/Tablets (Android/iOS).
- Optional AR headset (e.g., Hololens, Magic Leap – if available).

Software:

- **AR Development Platform:** Unity 3D with Vuforia Engine or 8thWall (WebAR) / ZapWorks.
- **3D Modeling Tools:** Blender / SolidWorks / Fusion 360 (to export CAD as .fbx/.glb for Unity).
- **Programming Language:** C# (for Unity) or JavaScript (for WebAR).
- **QR Code Tools:** Embedded QR codes for station-linked AR triggers.
- **Optional CMS/Database:** Firebase or Google Sheets for dynamically updating content.

Key Deliverables:

1. **QR Code Integration:**
   - Unique QR code for **[Station Name]** that launches AR experience on device.
2. **3D Visualization of Station:**
   - CAD-to-AR conversion of the machine or setup with visual fidelity.
   - Annotated 3D model showing parts, functionality, and interactions.
3. **Interactive AR Content:**
   - Text, audio, or video descriptions.
   - Process animations or guided step-by-step sequences (e.g., material flow, inspection method).
4. **Cross-Platform AR Application:**
   - APK or WebAR link for Android and iOS deployment.
   - Simple, intuitive user interface for scan-trigger-view experience.
5. **Testing & Feedback:**
   - Usability testing with faculty/students.
   - Performance tuning for various mobile devices.
6. **Final Documentation:**
   - User manual for app operation.
   - Developer documentation including source code, assets, and build instructions.
   - Suggestions for future scalability (e.g., multilingual support, integration with live sensor data).

Project: 2025.06.11

Title of the Project:

"Development of Interactive Web Portal with Chatbot for the Center of Excellence in Digital Manufacturing"

Objective:

- To design and develop a dynamic and user-friendly website for the Center of Excellence in Digital Manufacturing that showcases detailed information about all stations (MIRAC-PC, TRIAC-PC, Assembly, Inspection, Performance Measurement, AS/RS), current and past projects, faculty and student involvement, and publications.
- To implement an AI-based chatbot that assists users in navigating the website and exploring the center's facilities, capabilities, and ongoing activities.
- To ensure the platform is responsive, informative, and scalable for future expansion and integration.

Present Status:

- The Center of Excellence infrastructure is being developed and various stations are at different stages of implementation.
- Project details, faculty/staff list, and student contributions are being documented.

Technologies to be Used:

Frontend:

- **Framework:** React.js / Next.js (preferred for modern UI and performance)
- **Styling:** Tailwind CSS / Bootstrap
- **UI Libraries:** ShadCN / Material UI / Chakra UI

Backend:

- **Server-side:** Node.js with Express or Firebase for serverless backend
- **Database:** Firebase Firestore / MongoDB / PostgreSQL (for dynamic content like projects, people, etc.)

Chatbot Integration:

- **Custom Chatbot:** Powered by OpenAI API or Dialogflow
- **NLP Platform:** Rasa (for fully open-source solution) or Microsoft Bot Framework

Content Management:

- Headless CMS (optional): Strapi / Sanity / Contentful for future content updates

Deployment:

- Vercel / Netlify for frontend
- Firebase / Render / Railway / DigitalOcean for backend (depending on architecture)

Key Deliverables:

1. **Interactive Website UI:**
    - Home page with center overview and highlights
    - Individual pages for each station (MIRAC-PC, TRIAC-PC, etc.) with 3D images or videos, features, and applications
    - Section for student and faculty involvement
    - Archive of projects, achievements, and publications
    - News and updates section (optional)
2. **Admin Panel or CMS (Optional):**
    - Interface for authorized users to add/edit projects, personnel, or updates
3. **Chatbot Integration:**
    - Conversational assistant to answer FAQs, explain station functionalities, help with navigation, and suggest relevant projects
    - Integration with website via floating widget
4. **Mobile Responsiveness:**
    - Fully optimized for smartphones and tablets
5. **Performance and SEO Optimization:**
    - Fast loading time, SEO tags for visibility, and structured content
6. **Documentation and Handover:**
    - Source code with documentation
    - Deployment and maintenance guide
    - Future improvement roadmap

Project: 2025.06.12

Title of the Project:

"Development of PC-Based Inspection Assistant for Geometric Measurement and IIoT Data Transfer in Manual Inspection Station"

Objective:

- To develop a Windows-based application that guides users through a step-by-step process of manually measuring geometric dimensions of machined components using digital instruments (e.g., Vernier Caliper).
- To automatically acquire measurement data from the instrument and transmit it to an IIoT platform (ThingWorx) via the connected PC.
- To ensure the system provides visual feedback, stores data locally, and maintains accuracy and traceability of measurements.

Present Status:

- Manual Inspection Station is under development.
- Measurement instruments (digital Vernier Caliper or similar) have been procured and can be interfaced with PC via USB/serial.
- ThingWorx IIoT platform is available and accessible.
- No application software for data capture, user guidance, or ThingWorx integration has been developed yet.

Technologies to be Used:

Hardware:

- Digital Vernier Caliper (with USB or serial interface).
- Windows PC as a local gateway.

Software:

- **Application Development:** Python (Tkinter/PyQt5) or C# (WPF/WinForms)
- **Serial Communication:** `pyserial` (Python) or equivalent in .NET
- **IIoT Integration:** ThingWorx REST API / MQTT / Kepware for data publishing
- **Data Storage (optional):** SQLite or CSV for local backup/logging

- **User Guidance UI:** Interactive GUI with part image, measurement sequence, and status indicator

---

Key Deliverables:

1. **PC-Based Measurement Assistant:**
    - GUI to guide the operator step-by-step on what to measure and how
    - Visual representation of component and measurement locations
2. **Live Data Capture:**
    - Automatic reading of dimensional values from digital Vernier via serial/USB
    - Validation and display of readings with unit conversion and timestamp
3. **IIoT Platform Integration:**
    - Real-time transfer of measurements to ThingWorx via REST API or Kepware bridge
    - Status update (e.g., pass/fail) and logging in ThingWorx dashboard
4. **Local Logging System (Optional):**
    - Backup of measurements in CSV or SQLite format with part ID, time, and values
5. **Testing and Validation:**
    - Test with sample parts and calibrate the system for accuracy
    - Confirm communication with ThingWorx and integrity of data transfer
6. **Final Documentation:**
    - Source code with user and developer manuals
    - Integration and setup guide
    - Performance report and improvement suggestions

Project: 2025.06.13

Title of the Project:

"AI-Based Vision System for Geometric Dimension Measurement of Machined Components"

Objective:

- To develop a vision-based application that can **automatically measure geometric dimensions** (length, width, diameter, etc.) of machined components using a standard camera.
- To apply **AI/ML techniques** to achieve robust measurements that are **independent of variations in camera angle, lighting conditions, and orientation**.
- To integrate the system with an IIoT platform (e.g., ThingWorx) for real-time data transfer and visualization.

Present Status:

- Concept finalized for vision-based measurement using AI/ML.
- Sample machined components are available for training/testing datasets.
- IIoT infrastructure (ThingWorx) is set up and accessible.
- No dataset, AI model, or image processing pipeline has been developed yet.
- Camera module to be finalized (e.g., industrial webcam or mobile device camera).

Technologies to be Used:

Hardware:

- High-resolution camera (USB camera or mobile camera)
- PC or Jetson Orin (for edge processing, if needed)

Software & Frameworks:

- **Programming Language:** Python
- **Computer Vision:** OpenCV, MediaPipe
- **AI/ML Frameworks:** TensorFlow, PyTorch (for object detection, pose estimation, etc.)
- **Pre-trained Models:** YOLOv8, Detectron2 (fine-tuned for dimension detection)
- **Angle and Light Invariance Techniques:**
    - Image augmentation during training

- Homography transformation
- Shadow removal and contrast normalization

- **IIoT Integration:** REST API or MQTT protocol to ThingWorx
- **Measurement Tools:** Perspective correction, calibration grid, or marker-based scaling

---

Key Deliverables:

1. **Dataset Preparation:**
   - Image dataset of various machined parts with ground truth dimensions.
   - Annotated images for training AI models (bounding boxes, keypoints).
2. **AI-Based Measurement Engine:**
   - Trained model for identifying key geometric features (e.g., edges, holes, faces).
   - Algorithms for converting pixel distances to real-world dimensions using camera calibration and reference markers.
3. **User Interface Application:**
   - Windows/Linux GUI to upload or capture live images and display measured values.
   - Visual feedback on detection accuracy and dimension lines.
4. **Robustness to Variability:**
   - Tolerance to changes in light, object rotation, and perspective distortion.
   - Test across multiple background conditions and part orientations.
5. **IIoT Platform Integration:**
   - Transfer of measured data to ThingWorx dashboard.
   - Logging of part ID, image, and measurement data.
6. **Testing & Validation:**
   - Accuracy evaluation against manual tools (Vernier Caliper).
   - Error percentage reporting and calibration tools.
7. **Final Documentation:**
   - Dataset structure, model training process, codebase.
   - Setup guide and user manual.
   - Performance evaluation report with improvement roadmap.

Project: 2025.06.14

Title of the Project:

“Simulation and Cyber-Physical System Emulation for the Center of Excellence in Digital Manufacturing using Emulate3D”

Objective:

1. To simulate the manufacturing facilities, including AS/RS, retrofitted MIRAC-PC and TRIAC-PC, manual inspection station, assembly station, and performance testing station, using Emulate3D software.
2. To develop a cyber-physical system (CPS) model by integrating hardware components with their virtual counterparts via protocols such as OPC-UA and Ethernet/IP.
3. To test and analyze the performance of the simulated manufacturing system, ensuring seamless communication between all stations and validating the efficiency of workflows.
4. To create a scalable framework for future enhancements and integration of additional systems.

Present Status:

1. **Infrastructure:**
    - **AS/RS:** Ready
    - **MIRAC-PC (Turning Center):** Retrofitting completed and functional
    - **TRIAC-PC (Machining Center):** Retrofitting 50% complete
    - **Assembly Station:** Ready
    - **Manual Inspection Station:** Under development
    - **Performance Testing Station:** Setup initiated
2. **Software:**
    - Emulate3D software procured and installed
    - OPC-UA and Ethernet/IP protocols identified for IIoT connectivity
3. **Connectivity:**
    - Integration of physical stations with Emulate3D CPS platform pending

---

Technologies to be Used:

1. **Software:**
    - **Emulate3D:** For virtual modeling and simulation of manufacturing workflows
    - **IIoT Platform (e.g., ThingWorx / Kepware):** For integration via OPC-UA and Ethernet/IP

- **Data Analysis Tools (e.g., Excel, Power BI, Python):** For evaluating performance metrics

2. **Hardware:**
    - AS/RS, MIRAC-PC, TRIAC-PC, Assembly Station, Manual Inspection Station, and Performance Testing Station
    - Industrial controllers, sensors, and network modules with IIoT capabilities
3. **Protocols:**
    - **OPC-UA:** For secure, standardized machine-to-machine communication
    - **Ethernet/IP:** For high-speed data exchange within industrial networks

---

Key Deliverables:

1. **3D Digital Twin Models:**
    - Accurate Emulate3D representations of all ready stations
    - Integration-ready model of TRIAC-PC (with pending components marked)
2. **Cyber-Physical Integration:**
    - Bidirectional data flow between Emulate3D and physical hardware using OPC-UA / Ethernet/IP
3. **Workflow Simulation:**
    - Visualization and validation of material flow across stations using simulated AMR path
4. **Performance Analysis Report:**
    - Evaluation of system performance: cycle time, bottlenecks, and station utilization
5. **Modular and Scalable Framework:**
    - Reusable architecture for future expansion (e.g., additional machines, inspection methods)
6. **Documentation:**
    - Project documentation covering simulation setup, integration steps, test results, and source files

---

Project: 2025.06.15

Title of the Project:

"Smart Conversion of a Conventional Lathe with IIoT Integration Using ThingWorx"

Objective:

1. To complete the **electronic and electrical connections** for integrating sensors, controllers, and other components.
2. To develop a **PLC program** for monitoring and controlling the operations of the smart lathe, including spindle speed, feed rate, and safety interlocks.
3. To establish IIoT connectivity for real-time data transfer to the **ThingWorx** platform using **OPC-UA** or **Ethernet/IP** protocols.
4. To design and develop **IIoT dashboards** on ThingWorx to display real-time machining data, performance metrics, and operational analytics.
5. To test and validate the smart lathe's functionality, ensuring seamless integration of electronics, PLC, and IIoT systems.

Present Status:

1. **Mechanical Changes**:
   - All necessary mechanical modifications have been completed.
2. **Electronics and Sensors**:
   - Electronics, sensors, and other components for data acquisition have been **procured**.
3. **Control Panel**:
   - The control panel for the smart lathe is **ready** and awaiting connection.
4. **Pending Tasks**:
   - Completion of **electronic/electrical connections** between sensors, motors, and PLC.
   - Development and testing of the **PLC program** for machine operations.
   - Establishing IIoT connectivity to the **ThingWorx platform**.
   - Development of **IIoT dashboards** for real-time monitoring and performance analysis.

Technologies to be Used:

Hardware:

- Sensors: For spindle speed, temperature, vibration, and tool position monitoring.
- PLC (Programmable Logic Controller): For machine control and operation.

- Control Panel: For electrical integration and safety.

Software:

- **PLC Programming**: To monitor and control the lathe's operations.
- **ThingWorx Platform**: For IIoT connectivity, real-time data transfer, and dashboard development.
- Communication Protocols: **OPC-UA** and **Ethernet/IP** for machine-to-cloud connectivity.

Data Visualization and Analytics:

- ThingWorx Dashboard: For real-time monitoring of machine status, energy consumption, operational efficiency, and fault diagnostics.

Key Deliverables:

1. **Electrical and Electronic Integration**:
   - Complete wiring and integration of electronics, PLC, and sensors with the smart lathe.
2. **PLC Program Development**:
   - Functional PLC program to monitor and control lathe parameters, including spindle speed, feed rate, and alarms.
3. **IIoT Connectivity**:
   - Real-time machine data transfer to **ThingWorx** using OPC-UA/Ethernet-IP protocols.
4. **ThingWorx Dashboards**:
   - Development of individual and combined dashboards for monitoring:
     - Real-time machining parameters (speed, feed rate, tool wear).
     - System health and fault diagnostics.
     - Machine utilization and efficiency analytics.
5. **System Testing and Validation**:
   - Testing the smart lathe to ensure proper integration of mechanical, electrical, and IIoT systems.
   - Performance report detailing efficiency improvements and data insights.

Project: 2025.06.16

Title of the Project:

"IIoT Integration and PLC Programming for Hydraulic Assembly Station"

Objective:

1. To implement PLC programming for the control of a hydraulic assembly station using an already developed hydraulic circuit and control panel.
2. To establish IIoT connectivity of the station with the **ThingWorx** platform using **OPC-UA** for real-time monitoring and control.
3. To configure and test the communication between the PLC and the IIoT system to ensure seamless data flow and control feedback.
4. To validate the complete setup through real-time performance testing and data logging.

Present Status:

1. **Hydraulic Assembly Station:**
   - Hydraulic circuit is developed and tested.
   - Control panel for the station is fully assembled and ready for programming.
2. **IIoT Connectivity:**
   - OPC-UA-based communication is planned but not yet implemented.
   - ThingWorx platform is available for integration.

Technologies to be Used:

1. **Hardware:**
   - Hydraulic press with existing hydraulic circuit.
   - Control panel including PLC, relays, sensors, and actuators.
2. **Software and Protocols:**
   - **PLC Programming:** Ladder Logic (Siemens / Omron / Allen-Bradley depending on controller).
   - **IIoT Integration:** OPC-UA protocol for data communication.
   - **IIoT Platform:** ThingWorx for real-time data visualization and control.
   - **HMI Interface:** Optional for user interaction with the hydraulic system.
   - **Data Logging:** For operational status and performance analytics.

Key Deliverables:

1. **PLC-Controlled Hydraulic Station:**
    - Functional automation of the hydraulic press via PLC programming.
2. **OPC-UA Connectivity:**
    - Successfully established data link between PLC and ThingWorx.
3. **Real-Time Monitoring Dashboard:**
    - Visualization of system parameters (pressure, status, cycle count) on ThingWorx.
4. **Tested and Validated System:**
    - Performance validation report covering system behavior under different scenarios.
5. **Documentation:**
    - PLC logic diagrams, OPC-UA setup guide, ThingWorx configuration steps, and final project report.
