Protonetix Private Limited – Complete Company Overview

Introduction

Protonetix Private Limited is a technology-driven company focused on **robotics, automation, artificial intelligence (AI), the Internet of Things (IoT), embedded systems, drone technology, and programming**. Our goal is to **redefine innovation** by developing cutting-edge **robotics solutions, training the next generation of tech leaders, and offering industry-grade automation services**.

We operate in **four key divisions**:

- **Research & Development (R&D)** – Developing advanced robotic systems, AI models, automation solutions, and IoT applications.
- **Edutech & Training** – Offering **industry-oriented** courses in **robotics, AI, IoT, full-stack development, and drone technology**.
- **Industrial Solutions & Consulting** – Helping businesses adopt **automation, robotics, AI, and IoT for Industry 4.0**.
- **Programming & Placement Training** – Preparing students and professionals for tech careers through **hands-on programming, data structures & algorithms (DSA), interview preparation, and competitive coding**.

1. Research & Development

(R&D) Division

Our R&D team works on **real-world problems** to create **innovative, AI-driven solutions** in robotics, automation, IoT, and embedded systems.

A. Robotics & Automation

- **Autonomous Mobile Robots (AMRs) & Drones**
- SLAM-based navigation (ROS & ROS 2)
- AI-powered object detection & avoidance
- Drone path planning & real-time mapping
- **Humanoid & Service Robots**
- AI-driven assistants for home & industrial use
- Emotion & voice recognition systems
- **AI-Powered Industrial Automation**

- Smart robotic arms for manufacturing
- AI-based quality inspection systems

B. AI & Machine Learning

- **Computer Vision Applications**
- Object detection & recognition (YOLO, OpenCV)
- Pose estimation & OCR
- AI-powered surveillance systems
- **NLP-based AI Assistants**
- Chatbots for business automation
- Speech-to-text & text-to-speech conversion
- **AI-Driven Predictive Maintenance**
- Machine health monitoring in industries
- AI-based anomaly detection in industrial equipment

C. IoT & Embedded Systems

- **Smart Home & Industrial IoT Solutions**
- IoT-based water flow monitoring & smart energy management
- Cloud-integrated IoT applications
- **Custom PCB Design & Embedded Development**
- ESP32 & Raspberry Pi-based automation projects

D. Drone Technology & UAV Development

- **Indoor & Outdoor Autonomous Navigation**
- LiDAR-based mapping & SLAM
- AI-driven obstacle detection & avoidance
- **Surveillance & Security Drones**
- High-resolution imaging for aerial

surveillance
- AI-powered real-time tracking
-

2. Edutech & Training Division

We offer **industry-focused training programs** designed for students, professionals, and enterprises. Our courses emphasize **hands-on experience, real-world projects, and industry-ready skills**.

Training Programs:

- **Robotics & Automation Engineering**
- **AI & Machine Learning**
- **IoT & Embedded Systems**
- **Drone Technology & UAV Development**
- **Full-Stack Web Development**

- **Programming & Placement Training (DSA, Competitive Coding, Interview Prep)**

3. Industrial Solutions & Consulting Division

We help **startups, enterprises, and industries** implement **automation, robotics, AI, and IoT** to enhance productivity and efficiency.

Our Industrial Solutions:

- **Smart Factory & Industry 4.0**
- AI-based predictive maintenance
- Industrial IoT for real-time monitoring
- **Autonomous Warehouse Management**
- AI-driven robotic automation

- Smart inventory tracking systems
- **AI-powered Predictive Analytics**
- Data-driven decision-making
- Process optimization using AI

4. Programming & Placement Training Division

This division **prepares students and professionals for top tech careers** with expert-led training in programming, DSA, full-stack development, and interview preparation.

Key Offerings:

- **Data Structures & Algorithms (DSA)**
- **Competitive Programming (Codeforces, LeetCode, etc.)**

- **Full-Stack Web Development (Next.js, Python, MongoDB, REST APIs)**
- **Placement Preparation (Coding Rounds, System Design, Resume Building)**
- **Mock Interviews & Internship Guidance**

Boss, here’s a detailed breakdown for two separate curricula—one for ROS1 and one for ROS2—designed specifically for degree students. Both programs start with the basics and gradually build to more advanced robotics and automation topics using our ProtoBot platform. The idea is to make each program accessible while covering real-world concepts step by step.

Protonetix ProtoBot Robotics & Automation Engineering Training (ROS1)

Overview

This ROS1 program is tailored for students who are new to robotics and wish to learn the fundamentals using ROS1 (e.g., ROS Noetic). It covers basic hardware integration, ROS concepts, sensor interfacing, navigation, and project-based learning—all using ProtoBot.

Prerequisites

- Basic familiarity with Python or C++ (helpful but not mandatory)
- Basic knowledge of Linux and command-line usage
- A PC running Ubuntu (preferably 20.04 for ROS Noetic) with ROS1 installed

Curriculum Structure

[Lab 1] – Introduction to ProtoBot Hardware & ROS1 Basics

- **Session 1:**
- Overview of ProtoBot’s hardware components (mini CPU, microcontroller, motor drivers, sensors such as LIDAR, ultrasonic, IMU, and encoders)
- Understanding the role of each component in a robotics system
- **Session 2:**
- Introduction to ROS1: What it is and why we use it
- Basic ROS concepts: Nodes, Topics, and Messages
- Hands-on: Starting the ROS master (roscore) and exploring basic command-line tools

[Lab 2] – ROS1 Workspace Setup & Package Creation

- **Session 1:**
- Setting up a ROS workspace
- Creating your first ROS package
- Overview of ROS package structure and dependency management
- **Session 2:**
- Hands-on: Writing a simple publisher and subscriber in Python
- Testing communication between nodes

[Lab 3] – Basic ROS1 Communication and Debugging

- **Session 1:**
- Deep dive into ROS topics: Publishing, subscribing, and message types
- Using ROS command-line tools for debugging (rostopic, rosnode, rqt)
- **Session 2:**
- Hands-on: Develop a simple sensor data simulator and visualize data

[Lab 4] – Launch Files & Multi-Node Coordination

- **Session 1:**
- Introduction to roslaunch and XML launch files
- Grouping multiple nodes and setting parameters
- **Session 2:**
- Hands-on: Create launch files to run your publisher and subscriber together

[Lab 5] – Sensor Integration: LIDAR, Ultrasonic, & IMU

- **Session 1:**
- How to interface real sensors with ProtoBot
- Reading and processing sensor data in ROS1
- **Session 2:**
- Hands-on: Integrate LIDAR and ultrasonic sensor data for obstacle detection

[Lab 6] – Visualization Tools in ROS1

- **Session 1:**
- Introduction to rviz and rqt tools for data visualization
- Configuring displays and monitoring sensor data
- **Session 2:**
- Hands-on: Visualize sensor outputs from ProtoBot and debug sensor issues

[Lab 7] – ROS1 Services & Actions

- **Session 1:**
- Concepts of ROS services and actions: Request/reply and long-running tasks
- Writing a simple service server and client
- **Session 2:**
- Hands-on: Develop a service to control ProtoBot’s motors and an action to perform a timed task

[Lab 8] – Odometry & Basic Navigation

- **Session 1:**

- Understanding odometry: Wheel encoders and pose estimation
- Calculating robot movement and orientation
- **Session 2:**
- Hands-on: Implement a node that publishes odometry data and control robot motion accordingly

[Lab 9] – SLAM & Mapping

- **Session 1:**
- Introduction to SLAM (Simultaneous Localization And Mapping) in ROS1
- Overview of algorithms such as GMapping
- **Session 2:**
- Hands-on: Create a simple map using ProtoBot's LIDAR data in a simulated environment

[Lab 10] – Autonomous Navigation & Path Planning

- **Session 1:**

- Introduction to the ROS Navigation Stack
- Path planning concepts (A*, Dijkstra's algorithm)
- **Session 2:**
- Hands-on: Configure the navigation stack and test autonomous movement on ProtoBot

[Lab 11] – Computer Vision Integration

- **Session 1:**
- Integrating a camera with ProtoBot
- Basics of image processing using OpenCV in ROS1
- **Session 2:**
- Hands-on: Write a node to capture and display camera frames, with a simple object detection demo

Final Open-Ended Project

- Form teams to design and implement an autonomous robotics solution using

ProtoBot
- Projects can include tasks like floor cleaning, delivery simulation, or surveillance
- Emphasis on applying learned concepts in a real-world scenario

Protonetix ProtoBot Robotics & Automation Engineering Training (ROS2)

Overview

The ROS2 program is designed to bring students up to speed with the latest in robotics middleware. It builds on the ROS concepts with enhanced features like improved communication and real-time capabilities. This curriculum uses ROS2 (e.g., ROS2 Humble) on ProtoBot to

develop modern robotics applications.

Prerequisites

- Basic programming skills (Python/C++)
- Familiarity with Linux (Ubuntu 22.04 recommended)
- Basic electronics knowledge
- A PC with ROS2 Humble installed

Curriculum Structure

[Lab 1] – Getting Started with ProtoBot & ROS2 Fundamentals

- **Session 1:**
- Introduction to ProtoBot’s hardware and its components
- Overview of ROS2: What’s new compared to ROS1 and its advantages
- **Session 2:**
- Hands-on: Starting a ROS2 environment (using ros2 launch and ros2 run)
- Explore the ROS2 command-line tools

[Lab 2] – Setting Up ROS2 Workspace & Package Creation

- **Session 1:**
- Creating and configuring a ROS2 workspace
- Building your first ROS2 package using colcon
- **Session 2:**
- Hands-on: Writing a simple ROS2 node in Python for ProtoBot
- Testing communication between nodes using ROS2 topics

[Lab 3] – ROS2 Communication: Publishers & Subscribers

- **Session 1:**
- Deep dive into ROS2 communication mechanisms: Publishers, Subscribers, and Message Types
- Comparison with ROS1 communication paradigms
- **Session 2:**

- Hands-on: Develop and run simple nodes that exchange sensor data

[Lab 4] – Launch Files & Parameter Handling in ROS2

- **Session 1:**
- Introduction to ROS2 launch system and Python-based launch files
- Configuring parameters and dynamic reconfiguration
- **Session 2:**
- Hands-on: Create launch files to start multiple nodes on ProtoBot

[Lab 5] – Sensor Integration & Data Visualization

- **Session 1:**
- Integrating ProtoBot's sensors (LIDAR, IMU, ultrasonic) in ROS2
- How to process sensor data and publish it as ROS2 messages
- **Session 2:**
- Hands-on: Visualize sensor data using

rviz2 and rqt in ROS2

[Lab 6] – ROS2 Services, Actions, and Real-Time Tasks

- **Session 1:**
- Concepts of ROS2 services and actions for managing long-running tasks
- Developing service servers and action clients in ROS2
- **Session 2:**
- Hands-on: Build a service to control ProtoBot's motor functions and an action for time-bound tasks

[Lab 7] – Teleoperation & Differential Drive Control

- **Session 1:**
- Introduction to teleoperation using ROS2 (teleop_twist_keyboard or joystick interfaces)
- Differential drive kinematics for ProtoBot
- **Session 2:**

- Hands-on: Implement teleoperation control nodes to manually drive ProtoBot

[Lab 8] – Advanced Sensor Integration: LiDAR & Obstacle Avoidance

- **Session 1:**
- Processing LiDAR data in ROS2 for obstacle detection
- Tuning sensor parameters for reliable data
- **Session 2:**
- Hands-on: Develop an obstacle avoidance algorithm and test it on ProtoBot

[Lab 9] – SLAM and Mapping in ROS2

- **Session 1:**
- Introduction to SLAM in ROS2 (using Hector SLAM or Cartographer)
- Understanding the mapping process and localization
- **Session 2:**

- Hands-on: Generate a map using ProtoBot’s sensor data in a simulated or controlled environment

[Lab 10] – Autonomous Navigation & Path Planning

- **Session 1:**
- Overview of the ROS2 Navigation Stack
- Path planning algorithms and setting up navigation parameters
- **Session 2:**
- Hands-on: Configure and test autonomous navigation on ProtoBot using the ROS2 navigation stack

[Lab 11] – Computer Vision and AI Integration

- **Session 1:**
- Integrating a camera with ProtoBot in ROS2
- Basics of computer vision using OpenCV and TensorFlow for simple object detection

- **Session 2:**
- Hands-on: Write a ROS2 node to process camera input and execute vision-based tasks

Final Open-Ended Project

- Form project teams and design an innovative ProtoBot application that integrates navigation, sensor data processing, and AI
- Example projects: Autonomous delivery robot, surveillance system, or a collaborative multi-robot task
- Emphasis on applying all learned ROS2 concepts in a real-world scenario

Iot Training Program.

Part I: Basics

1. Introduction to IoT & Electronics Fundamentals

- **Overview of IoT:**
- What is IoT? Applications in smart homes, industrial monitoring, and automation.
- **Basic Electronics:**
- Voltage, current, resistance, and Ohm's law
- Digital vs. analog signals
- Introduction to circuit design and breadboarding

2. IoT Boards & Platforms

- **Popular Boards:**
- Arduino (Uno, Nano)
- Raspberry Pi fundamentals
- Introduction to ESP32 and similar microcontrollers
- **Setting Up the Development**

Environment:

- Installing Arduino IDE, Raspberry Pi OS, or PlatformIO
- Basic programming concepts in C/C++ or Python

3. Introduction to Sensors and Actuators

- **Sensors:**
- Temperature, humidity, light, and motion sensors
- How sensors convert physical parameters to electrical signals
- **Actuators:**
- Basic types: LEDs, buzzers, DC motors, and servos
- Understanding how actuators perform physical actions

4. Hands-on Projects (Basics)

- Blinking an LED and controlling a buzzer
- Reading sensor data from a

temperature or light sensor
- Simple interaction: using a button to trigger an actuator (e.g., LED or motor)

Part II: Intermediate

1. Sensor Integration & Data Acquisition

- **Advanced Sensor Types:**
- Ultrasonic sensors for distance measurement
- Gas, pressure, and environmental sensors
- Basic sensor calibration and data smoothing
- **Interfacing Techniques:**
- Analog-to-digital conversion (ADC)
- Using I2C, SPI, and UART communication protocols

2. Enhanced Actuator Control

- **Advanced Actuators:**
- Servo motor control for precise movement
- Stepper motors and basic motor driver interfacing
- Using relays for switching higher loads
- **Control Methods:**
- Pulse-width modulation (PWM)
- Understanding motor driver circuits

3. Communication & Connectivity

- **Wired and Wireless Protocols:**
- Serial communication basics
- Introduction to Wi-Fi and Bluetooth connectivity with IoT boards
- Setting up simple wireless communication between devices

4. Intermediate Projects

- Home automation prototype: controlling lights and fans via sensor

inputs
- Environmental monitoring system: reading multiple sensor data and displaying it on an LCD or sending data to a PC
- Building a small sensor network using Arduino/ESP32 and basic wireless modules

Part III: Advanced

1. IoT System Architecture & Cloud Integration

- **IoT System Design:**
- Overview of IoT architectures (edge, fog, and cloud)
- Data collection, storage, and real-time processing
- **Cloud Connectivity:**

- Integrating IoT devices with cloud platforms (AWS IoT, Google Cloud IoT, or Azure IoT)
- Protocols like MQTT, HTTP, and CoAP for data transmission
- Securing data transmissions with SSL/TLS

2. Advanced Sensor Fusion & Actuator Coordination

- **Sensor Fusion Techniques:**
- Combining data from multiple sensors to improve accuracy
- Using filtering algorithms (e.g., Kalman filter) for sensor data smoothing
- **Advanced Actuator Control:**
- Coordinated control of multiple actuators in robotics and automation
- Feedback control systems and PID controllers

3. System Development and

Troubleshooting

- **Designing Complete IoT Systems:**
- End-to-end IoT system design, including hardware, firmware, and software
- Case studies on smart home or industrial IoT solutions
- **Security & Maintenance:**
- Implementing security best practices for IoT devices
- Regular maintenance, remote monitoring, and troubleshooting techniques

4. Advanced Projects

- Developing a smart home system with remote monitoring and control via a mobile app
- Industrial IoT monitoring system: integrating multiple sensors, real-time data analytics, and automated actuator

responses
- Building a fully functional, secure IoT prototype that incorporates cloud data processing, sensor fusion, and advanced control systems

Basics: Normal RC Controlled Drone

1. Introduction & Fundamentals

- **Overview of Drone Technology:**
- History and applications of RC drones
- Basic aerodynamics: lift, thrust, drag, and weight
- **Components Overview:**
- Frames, motors, propellers, Electronic Speed Controllers (ESCs)

- Basic flight controller functionality
- Radio Transmitters & Receivers for RC control

2. Assembly & Hardware Setup

- **Drone Assembly:**
- Step-by-step build of a basic RC drone kit
- Wiring and mounting components securely
- **Calibration & Testing:**
- Calibrating ESCs and radio controls
- Pre-flight safety checks and sensor calibration (IMU, gyroscope)

3. Manual Flight Control

- **RC Operation:**
- Fundamentals of manual control using a standard RC transmitter
- Understanding throttle, pitch, roll, and yaw
- **Basic Flight Maneuvers:**

- Hovering, takeoff, landing, and simple directional control
- Hands-on practice in controlled environments (open fields or flight simulators)

4. Basic Telemetry & Troubleshooting

- **Telemetry Basics:**
- Monitoring battery levels, signal strength, and flight parameters
- **Troubleshooting:**
- Identifying and resolving common issues (calibration errors, motor misalignment)

5. Hands-On Projects (Basics)

- **Project 1:** Build and fly a basic RC drone
- **Project 2:** Conduct manual flight tests and document performance
- **Project 3:** Implement simple troubleshooting steps based on real

flight data

Advanced: Mavlink and MAVROS Integration

1. Introduction to Advanced Communication Protocols

- **Mavlink Overview:**
- What is Mavlink and why it’s essential for advanced UAVs
- Comparison with traditional RC control: benefits for automation and telemetry
- **MAVROS Overview:**
- Integration of MAVROS with ROS for advanced control and data handling
- Overview of how MAVROS bridges ROS and MAVLink-enabled autopilots

2. Setting Up the Advanced

Environment

- **Hardware and Software Requirements:**
- Compatible autopilots (Pixhawk, APM) supporting MAVLink
- Installing ROS (ROS Noetic or ROS2, depending on your setup) and MAVROS
- **Configuration:**
- Setting up the communication link between the autopilot and the onboard computer
- Configuring parameters for reliable Mavlink messaging

3. Advanced Flight Control & Mission Planning

- **Autonomous Control:**
- Commanding the drone using MAVLink messages
- Setting up waypoint missions and autonomous flight routines
- **MAVROS Nodes & Topics:**

- Understanding key topics (e.g., /mavros/state, /mavros/setpoint_position/local)
- Creating custom ROS nodes to interface with MAVROS for mission control

4. Data Integration & Telemetry

- **Enhanced Telemetry:**
- Receiving and processing flight data (GPS, IMU, battery, etc.) via MAVLink
- Logging and visualizing flight data in real-time using ROS tools (rviz, rqt)
- **Real-Time Monitoring:**
- Implementing failsafe mechanisms and emergency protocols
- Integrating sensor fusion for improved navigation and obstacle avoidance

5. Hands-On Projects (Advanced)

- **Project 1:** Configure and test a Mavlink-enabled autopilot with MAVROS

- Setup a simulated mission and monitor the state using ROS
- **Project 2:** Develop an autonomous flight mission with waypoints and live telemetry
- Program a custom ROS node to send setpoints and adjust mission parameters on the fly
- **Project 3:** Integrate advanced sensor data (e.g., vision or LIDAR) with MAVROS
- Use sensor fusion to enhance the reliability of autonomous flight operations

Front end web development

Module 1: HTML & CSS Fundamentals

Topics Covered:

- **HTML Basics:**
- Document structure, semantic tags, forms, tables, and multimedia elements.
- **CSS Fundamentals:**
- Selectors, specificity, box model, and basic styling.
- **Responsive Design Principles:**
- Media queries, flexible layouts, and fluid images.
- **Hands-on Projects:**
- Build static web pages to practice semantic markup and layout techniques.

Module 2: Advanced CSS & UI/UX Design

Topics Covered:

- **Advanced Styling Techniques:**
- Flexbox, CSS Grid, pseudo-classes/ elements, and custom properties (CSS variables).
- **CSS Animations & Transitions:**
- Keyframes, animation timing, and interaction effects.
- **Pure CSS Component Design:**
- Creating reusable UI components (buttons, cards, navigation bars) without external libraries.
- **UI/UX Principles:**
- Color theory, typography, whitespace, and visual hierarchy.
- **Hands-on Projects:**
- Design and implement a fully responsive website layout using only pure CSS.

Module 3: JavaScript Fundamentals

Topics Covered:

- **Core JavaScript Concepts:**
- Variables, data types, control structures, functions, and error handling.
- **DOM Manipulation & Event Handling:**
- Selecting, modifying elements, and managing user interactions.
- **Asynchronous Programming:**
- Callbacks, Promises, and Fetch API for data retrieval.
- **Hands-on Projects:**
- Create interactive UI elements like modals, image sliders, and dynamic forms.

Module 4: Modern JavaScript & Frontend Frameworks

Topics Covered:

- **ES6+ Features:**

- Arrow functions, template literals, destructuring, spread/rest operators.
- **Introduction to React & Next.js:**
- Component-based architecture, state management, props, and lifecycle methods.
- **TypeScript Integration (Optional but Recommended):**
- Adding static type checking to your projects.
- **Server-Side Rendering (SSR) with Next.js:**
- Routing, data fetching, and SEO benefits.
- **Hands-on Projects:**
- Build a simple Next.js application with dynamic pages and basic API integration.

Module 5: Frontend Tooling & Best Practices

Topics Covered:

- **Version Control:**
- Git fundamentals and collaborative workflows.
- **Code Quality Tools:**
- Setting up ESLint, Prettier, and Git hooks.
- **Build Tools & Bundlers:**
- Introduction to Webpack/Vite for optimized asset management.
- **SEO & Accessibility:**
- Best practices for metadata, structured data, and WCAG guidelines.
- **Hands-on Projects:**
- Refactor an existing project to meet coding standards and optimize performance.

Module 6: Advanced Frontend Topics & Performance

Optimization

Topics Covered:

- **Progressive Web Apps (PWAs):**
- Service workers, offline capabilities, and caching strategies.
- **Performance Optimization:**
- Code splitting, lazy loading, and critical rendering paths.
- **Advanced Animations & Interactivity:**
- Using CSS and JavaScript libraries to create rich interactive experiences.
- **Hands-on Projects:**
- Enhance a web application with advanced performance techniques and smooth animations.

Capstone Project

Project Brief:

- **Objective:**
- Build a complete, professional website

for a fictional company that embodies strong UI/UX, responsive design, and high performance.

- **Requirements:**
- Use pure HTML, CSS, and JavaScript (with React/Next.js for dynamic content).
- Optimize for SEO, accessibility, and mobile responsiveness.
- Deploy the website on a live server with version control and continuous integration.
- **Outcome:**
- A portfolio-worthy website demonstrating comprehensive frontend skills and attention to detail.

Python development

Part I: Python Basics

Module 1: Introduction to Python

- **Overview & Setup:**
- What is Python? Its advantages and use cases.
- Installing Python and setting up a development environment (IDEs, editors).
- Using the interactive shell and running Python scripts.
- **Fundamental Concepts:**
- Variables, data types, basic operators.
- Input/Output operations and simple string manipulations.

Module 2: Control Structures and Data Structures

- **Control Structures:**
- Conditional statements (if, elif, else).
- Looping constructs (for, while) and comprehension syntax.
- **Core Data Structures:**

- Lists, tuples, dictionaries, and sets.
- Basic operations on these structures (iteration, slicing, and methods).

Module 3: Functions and Modules

- **Functions:**
- Defining and calling functions, scope and lifetime of variables.
- Lambda functions and higher-order functions (map, filter, reduce).
- **Modules & Packages:**
- Importing standard libraries and third-party modules.
- Creating and managing your own modules and packages.

Module 4: File I/O and Exception Handling

- **File Operations:**
- Reading from and writing to files.
- Working with different file types (text,

CSV, JSON).
- **Error Handling:**
- Using try, except, finally blocks.
- Creating custom exceptions and debugging strategies.

Module 5: Introduction to Object-Oriented Programming (OOP)

- **OOP Fundamentals:**
- Classes and objects, attributes, and methods.
- Inheritance, polymorphism, encapsulation, and abstraction.
- **Practical OOP:**
- Building simple class hierarchies and using built-in OOP features.

Module 6: Hands-On Projects and Mini Challenges

- **Mini Projects:**
- Command-line calculator, simple data parser, or file organizer.

- Practice with debugging and writing unit tests.
- **Capstone for Basics:**
- Develop a small, modular application (e.g., a contact management system) to apply the fundamentals.

Part II: Advanced Python – Computer Vision and Deep Learning

Module 1: Advanced Python Libraries and Data Manipulation

- **NumPy & Pandas:**
- Array operations, vectorized computations, and data manipulation.
- Data cleaning, filtering, and analysis with Pandas.
- **Matplotlib & Seaborn:**
- Data visualization for exploratory data analysis.

Module 2: Introduction to Computer Vision with OpenCV

- **OpenCV Basics:**
- Installing OpenCV and reading image/ video files.
- Basic image processing: resizing, cropping, color space conversions.
- **Image Processing Techniques:**
- Filtering, edge detection, and morphological operations.
- Contour detection, feature extraction, and transformations.
- **Hands-On Projects:**
- Create a photo editor tool, motion detection, or a simple image filter application.

Module 3: Deep Learning Foundations with TensorFlow (and Keras)

- **TensorFlow Basics:**
- Overview of TensorFlow and Keras API.
- Building and training a simple neural network.
- **Data Preparation:**
- Preprocessing data, working with image datasets.
- Data augmentation techniques.
- **Model Training & Evaluation:**
- Implementing training loops, loss functions, and optimization.
- Evaluating model performance and tuning hyperparameters.
- **Hands-On Projects:**
- Develop a digit recognizer (MNIST) or a simple image classifier.

Module 4: Advanced Deep Learning & Computer Vision Applications

- **Convolutional Neural Networks (CNNs):**

- Understanding CNN architecture and layers.
- Building CNN models for image classification, object detection, or segmentation.
- **Integrating OpenCV and TensorFlow:**
- Real-time video processing and object detection.
- Using pre-trained models and transfer learning.
- **Advanced Projects:**
- Develop an object detection system, facial recognition, or a real-time video processing application.

Module 5: Capstone Project – End-to-End AI Application

- **Project Overview:**
- Choose a real-world problem, such as automated quality control in manufacturing or a smart surveillance

system.

- **Implementation:**
- Integrate data collection, advanced image processing with OpenCV, and deep learning using TensorFlow.
- Build, train, and deploy the model, then integrate with a user interface if needed.
- **Evaluation & Iteration:**
- Test, refine, and optimize the system based on performance metrics.

Mechanical and electronics

PCB Design Training Program

Module 1: Introduction to CAD & Design Principles

Topics Covered:

- **Overview of CAD Tools:**
- What is CAD? Differences between 2D sketching and 3D modeling.
- Introduction to design standards, tolerances, and best practices.
- **Fundamental Design Principles:**
- Basic geometry, dimensioning, and drafting.
- Understanding material properties and production considerations.
- **Hands-on Activity:**
- Create simple sketches and 2D drawings using a basic CAD tool or paper prototypes.

Module 2: Mechanical Design with CATIA

Topics Covered:

- **CATIA Fundamentals:**
- Overview of CATIA’s interface and modules.

- Creating sketches, extrusions, and basic 3D models.
- **3D Modeling Techniques:**
- Part design: features, fillets, chamfers, and patterns.
- Assembly design: managing multiple parts, constraints, and joints.
- **Simulation & Analysis:**
- Introduction to basic stress analysis and design validation within CATIA.
- **Hands-on Activity:**
- Model a simple mechanical component (e.g., a bracket or gear) and assemble a small mechanism.

Module 3: Mechanical Design with Fusion 360

Topics Covered:

- **Fusion 360 Basics:**
- Interface tour, sketching, and parametric modeling.

- Difference between parametric and direct modeling.
- **Advanced Modeling Techniques:**
- Sculpting, surface modeling, and organic shapes.
- Creating assemblies and using simulation tools for motion and stress analysis.
- **Collaboration & Cloud Integration:**
- Managing versions, sharing designs, and cloud-based collaboration.
- **Hands-on Activity:**
- Design a small assembly (e.g., a custom enclosure or mechanical tool) and run a simulation for performance checks.

Module 4: PCB Design Fundamentals

Topics Covered:

- **Introduction to PCB Design:**

- Overview of PCB design process: schematic capture, layout, and routing.
- Understanding PCB layers, materials, and manufacturing considerations.
- **Schematic Design:**
- Component selection, symbol creation, and circuit connectivity.
- Using industry-standard PCB design software (e.g., KiCad, Eagle, or Altium Designer).
- **PCB Layout & Routing:**
- Best practices for component placement, trace routing, and signal integrity.
- Thermal management and EMI/EMC considerations.
- **Hands-on Activity:**
- Design a simple PCB for an embedded system (e.g., a sensor interface board) and prepare it for fabrication.

Module 5: Advanced Integrated Design

Topics Covered:

- **Integrated Mechanical & Electronics Design:**
- Designing mechanical enclosures that accommodate PCBs and other electronics.
- Techniques for mounting, wiring, and ensuring proper heat dissipation.
- **Advanced Simulation & Optimization:**
- Finite Element Analysis (FEA) for structural integrity in Fusion 360 or CATIA.
- Electromagnetic simulation for PCB performance.
- **Production & Prototyping:**
- Design for manufacturability (DFM) and rapid prototyping methods.
- Iterative design improvements based

on test feedback.

- **Hands-on Activity:**
- Develop an end-to-end project (e.g., a robotic sensor module or an IoT device) that integrates a custom-designed PCB with a mechanical enclosure. Prepare a complete design package ready for prototyping.

Module 6: Capstone Project

Project Brief:

- **Objective:**
- Execute a comprehensive design project that integrates mechanical and electronics design.
- **Project Examples:**
- Design a compact drone component, an IoT device enclosure with an integrated PCB, or a robotic arm mechanism.
- **Process:**

- From conceptual sketches and CAD modeling in CATIA/Fusion 360 to schematic capture and PCB layout, document the entire design lifecycle.
- **Evaluation:**
- Focus on design accuracy, innovation, and adherence to industry best practices.

Placement Focused Training Program

Module 1: Fundamentals of Data Structures & Algorithms

- **Core Concepts:**
- Arrays, Linked Lists, Stacks, Queues
- Trees (Binary, BST, AVL), Graphs, Hash Tables

- Sorting (Merge, Quick, Heap) and Searching Algorithms
- **Problem-Solving Basics:**
- Recursion and Backtracking
- Dynamic Programming and Greedy Algorithms
- **Hands-On:**
- Implement data structures in your preferred language
- Solve practice problems on platforms like LeetCode or HackerRank

Module 2: Problem-Solving & Competitive Programming

- **Pattern Recognition:**
- Common problem patterns and coding techniques
- Sliding window, two pointers, and divide & conquer strategies
- **Competitive Programming Practices:**
- Timed coding challenges and contests

- Optimization techniques and debugging strategies
- **Hands-On:**
- Participate in mock coding contests
- Regular practice sessions with progressively challenging problems

Module 3: Advanced Interview Topics & Coding Challenges

- **Deep Dive into Advanced Algorithms:**
- Graph algorithms (BFS, DFS, Dijkstra, etc.)
- Advanced Dynamic Programming and Bit Manipulation
- **Interview Question Bank:**
- Frequently asked questions in technical interviews
- Hands-on coding exercises with real-world scenarios
- **Hands-On:**
- Weekly coding sessions focused on

difficult problems
- Peer-to-peer coding sessions to discuss and optimize solutions

Module 4: System Design & Architecture Fundamentals

- **Introduction to System Design:**
- Basic principles of scalability, load balancing, and caching
- Designing distributed systems and microservices architecture
- **Key Concepts:**
- Database design (SQL/NoSQL), API design, and messaging queues
- Common design patterns and real-world case studies
- **Hands-On:**
- Design a scalable system (e.g., URL shortener, social media feed)
- Group projects to simulate system design discussions

Module 5: Soft Skills, Resume Building & Behavioral Preparation

- **Communication & Presentation:**
- Effective communication techniques for technical interviews
- Articulating thought processes clearly during problem solving
- **Resume & Personal Branding:**
- Crafting a standout technical resume
- Building a professional online portfolio (GitHub, LinkedIn)
- **Behavioral Interview Preparation:**
- Mock behavioral interviews and group discussions
- Handling situational and HR questions confidently
- **Hands-On:**
- Resume review workshops with industry experts
- Mock sessions with feedback on

communication and presentation skills

Module 6: Mock Interviews & Real-Time Practice Sessions

- **Technical Mock Interviews:**
- One-on-one sessions with industry professionals
- Simulated interviews covering coding, system design, and soft skills
- **Feedback & Iteration:**
- Detailed feedback and personalized improvement plans
- Regular practice sessions to track progress and build confidence
- **Hands-On:**
- Weekly mock interview rounds with timed challenges
- Live coding interviews and system design walkthroughs

Collaborations:

Newrro
Sri Sathya Sai Loka Seva Gurukulam
Sai Vidya Institute of Technology

Still more to come on the way....