

Government of Pakistan

**National Vocational and Technical Training Commission**  
**(NAVTTTC)**

"Prime Minister Youth Skill Development Program"



**Course Contents / Lesson Plan**

**Course Title:** Huawei HCCDA-AI

**Duration:** 3 Months

Author Name	Ahtisham Masood Consultant					
Course Title	Huawei HCCDA-AI					
Training Objective & Outcomes	Prepare candidates for the <b>Huawei HCCDA-AI certification</b> by covering AI/ML fundamentals, Huawei Cloud AI services, ModelArts, deep learning frameworks, and real-world AI application development.					
Entry-level of trainees	<ul style="list-style-type: none"><li>Currently enrolled in at least the 7th semester of a bachelor's degree program or pursuing a graduate/postgraduate degree in a relevant field (e.g., IT, Computer Science, Engineering).</li><li>Candidates with a completed Diploma in Information Technology (IT/DIT) are also eligible.</li></ul>					
Minimum Qualification of Teachers	<ul style="list-style-type: none"><li>Minimum Master’s degree (16 Years) in Computer Science, Information Technology, Software Engineering, or a closely related field from a recognized institution.</li><li>Huawei Certified (preferably HCCDA or higher) or equivalent certification in cloud technologies and AI.</li><li>At least 2 years of industry experience in cloud computing, AI, or related domains.</li><li>Strong knowledge of cloud infrastructure, virtualization, machine learning fundamentals, and hands-on lab environments.</li><li>Proven ability to deliver training, evaluate student progress, and support learners in a hands-on, project-based learning setup.</li><li>Experience with Huawei ModelArts, Mindspore or other Huawei AI tools.</li></ul>					
Scheme Studies of	Huawei HCCDA-AI (3-Month Course)					
	Sr. No	Main Topics	Theory Hrs.	Practical Hrs.	Total Hrs.	Credit Hours
	1.	Intro to AI & Huawei Strategy	10.5	4.5	15	
	2.	Python for AI (ModelArts tools)	6.25	8.75	15	
	3.	Huawei Cloud EI API Services	6.25	8.75	15	
	4.	Requirement Analysis for AI Integration.	7.75	7.25	15	
	5.	Huawei Cloud ModelArts Introduction	6.5	8.5	15	
	6.	HiLens Architecture & Use Cases	5.75	9.25	15	

	<b>7.</b>	<b>Deep Learning Concepts &amp; Frameworks</b>	<b>7.0</b>	<b>8.0</b>	<b>15</b>	
	<b>8.</b>	<b>Model Deployment &amp; Integration</b>	<b>6.5</b>	<b>8.5</b>	<b>15</b>	
	<b>9.</b>	<b>Testing &amp; Optimization for AI Apps</b>	<b>6.5</b>	<b>8.5</b>	<b>15</b>	
	<b>10.</b>	<b>MLOps &amp; CI/CD on Huawei Cloud</b>	<b>5.75</b>	<b>9.25</b>	<b>15</b>	
	<b>11.</b>	<b>Exam Preparation &amp; Certification Focus</b>	<b>6.75</b>	<b>8.25</b>	<b>15</b>	
	<b>12.</b>	<b>Capstone Project Development &amp; Review</b>	<b>3.0</b>	<b>12.0</b>	<b>15</b>	
	<b>Total</b>		<b>78.5 hrs</b>	<b>101.5 hrs</b>	<b>180 hrs</b>	
<b>Course Execution Plan</b>	Duration of the course: <b>3 months (12 Weeks)</b> Theory: <b>44.5%</b> Practical: <b>55.5%</b> Weekly hours: <b>15 hours per week</b> Total contact hours: <b>Maximum 180 hours</b>					
<b>Companies offering jobs in the respective trade</b>						
<b>No of Students</b>	25 - 35					
<b>Learning Place</b>	Classroom / Lab					
<b>Instructional Resources</b>	<ul style="list-style-type: none"> <li>- <b>Huawei Talent Online – LMS &amp; certification tracking</b></li> <li>- <b>ModelArts – AI development &amp; deployment</b></li> <li>- <b>MindSpore / TensorFlow – Deep learning frameworks</b></li> <li>- <b>Python (Jupyter/Colab) – Core programming</b></li> <li>- <b>Huawei Cloud – For cloud-AI integration labs</b></li> </ul>					

## DETAIL OF COURSE CONTENTS

Module Title	Learning Units	• Task/Practical
<b>Introduction to AI &amp; Huawei Strategy</b>	<b>Day 1</b> <ul style="list-style-type: none"> <li>What is AI? History, types, and scope of AI</li> </ul>	<ul style="list-style-type: none"> <li>Define AI and distinguish between weak, strong, and general AI</li> <li>Explore historical evolution of AI</li> <li>Discuss symbolic vs. machine learning AI</li> <li>Identify key domains (vision, NLP, robotics)</li> </ul>
	<b>Day 2</b> <ul style="list-style-type: none"> <li>AI industry ecosystem, trends, and challenges in adoption</li> </ul>	<ul style="list-style-type: none"> <li>Present global AI industry landscape (companies, sectors)</li> <li>Identify AI trends (e.g., GenAI, edge AI, LLMs)</li> <li>Discuss adoption barriers (data, regulation, cost)</li> <li>Use a Gartner Hype Cycle visual</li> </ul>
	<b>Day 3</b> <ul style="list-style-type: none"> <li>Huawei's AI strategy &amp; open AI capabilities from Huawei Cloud EI</li> </ul>	<ul style="list-style-type: none"> <li>Introduce Huawei Cloud EI portfolio (vision, NLP, speech, OCR)</li> <li>Discuss ModelArts, HiLens integration</li> <li>Showcase Huawei's global AI initiatives</li> <li>Map use cases to Huawei capabilities</li> </ul>
	<b>Day 4</b> <ul style="list-style-type: none"> <li>Overview of AI certification exam: domains, sample questions, and preparation tips</li> </ul>	<ul style="list-style-type: none"> <li>Review exam format and weightage by domain</li> <li>Analyze sample questions and answer structure</li> <li>Discuss learning strategy and revision plan</li> <li>Take a short pre-test (10–12 questions)</li> </ul>
	<b>Day 5</b> <ul style="list-style-type: none"> <li>Quiz + group discussion on AI trends &amp; Huawei strategy</li> </ul>	<ul style="list-style-type: none"> <li>Conduct a formal quiz on Weeks 1 content</li> <li>Review each answer with rationales</li> <li>Group debate: AI hype vs reality</li> <li>Trainer Q&amp;A on career scope in AI</li> </ul>
<b>Python for AI (ModelArts tools)</b>	<b>Day 6</b> <ul style="list-style-type: none"> <li>Python basics review: variables, loops, functions</li> </ul>	<ul style="list-style-type: none"> <li>Setup Python 3 environment on ModelArts</li> <li>Review variables, data types, and conditionals</li> <li>Practice writing functions and loops</li> <li>Implement simple logic-based examples</li> </ul>
	<b>Day 7</b> <ul style="list-style-type: none"> <li>NumPy arrays: creation, operations,</li> </ul>	<ul style="list-style-type: none"> <li>Import NumPy and create arrays from lists</li> </ul>

	broadcasting	<ul style="list-style-type: none"> <li>• Apply indexing, slicing, reshaping</li> <li>• Demonstrate broadcasting and matrix multiplication</li> <li>• Solve numerical problems with NumPy</li> </ul>
	<b>Day 8</b> <ul style="list-style-type: none"> <li>• Pandas for data manipulation + Scikit-learn basics</li> </ul>	<ul style="list-style-type: none"> <li>• Load datasets with Pandas (CSV/Excel)</li> <li>• Explore dataframes: filtering, grouping, missing values</li> <li>• Train a basic classifier with scikit-learn</li> <li>• Evaluate accuracy with simple metrics</li> </ul>
	<b>Day 9</b> <ul style="list-style-type: none"> <li>• Using ModelArts notebooks: environment setup, code, data access</li> </ul>	<ul style="list-style-type: none"> <li>• Launch a notebook on Huawei ModelArts</li> <li>• Upload and organize datasets on OBS</li> <li>• Mount OBS to the notebook environment</li> <li>• Run a test script using built-in tools</li> </ul>
	<b>Day 10</b> <ul style="list-style-type: none"> <li>• Practical: manipulate data using NumPy and Pandas in ModelArts</li> </ul>	<ul style="list-style-type: none"> <li>• Import real dataset (e.g., Titanic, Iris)</li> <li>• Clean and transform data with Pandas</li> <li>• Perform statistical analysis using NumPy</li> <li>• Visualize outputs using Matplotlib or Seaborn</li> </ul>
Huawei Cloud EI API Services	<b>Day 11</b> <ul style="list-style-type: none"> <li>• EI overview: image, NLP, OCR, and speech services overview</li> </ul>	<ul style="list-style-type: none"> <li>• Introduce Huawei EI platform and service categories</li> <li>• Explain key services: image tagging, speech-to-text, sentiment analysis, OCR</li> <li>• Identify application scenarios for each service</li> <li>• Review sample output and architecture diagrams</li> </ul>
	<b>Day 12</b> <ul style="list-style-type: none"> <li>• Authentication and API calling methods: REST, SDKs (Python)</li> </ul>	<ul style="list-style-type: none"> <li>• Set up Huawei Cloud account and create access keys</li> <li>• Install SDK and configure credentials securely</li> <li>• Practice REST calls using Postman and Python requests</li> <li>• Explore rate limits, auth headers, and status codes</li> </ul>
	<b>Day 13</b> <ul style="list-style-type: none"> <li>• Deep dive into General Table OCR API</li> </ul>	<ul style="list-style-type: none"> <li>• Explain use cases for table OCR (invoices, reports, receipts)</li> <li>• Understand API parameters and response structure</li> </ul>

		<ul style="list-style-type: none"> <li>• Use API docs to build request body</li> <li>• Review output JSON and extract cell values</li> </ul>
	<b>Day 14</b> <ul style="list-style-type: none"> <li>• Practical: Call and test General Table OCR API</li> </ul>	<ul style="list-style-type: none"> <li>• Upload a test table image to OBS</li> <li>• Write Python script to invoke OCR API with image link</li> <li>• Parse response and visualize the extracted data</li> <li>• Handle common errors (timeout, image type)</li> </ul>
	<b>Day 15</b> <ul style="list-style-type: none"> <li>• Practical: Evaluate and log API results</li> </ul>	<ul style="list-style-type: none"> <li>• Create multiple test cases with varied tables</li> <li>• Log response time and accuracy for each</li> <li>• Compare results with ground truth data</li> <li>• Generate a simple evaluation report (precision/recall summary)</li> </ul>
<b>Requirement Analysis for AI Integration</b>	<b>Day 16</b> <ul style="list-style-type: none"> <li>• Requirement analysis: techniques and templates</li> </ul>	<ul style="list-style-type: none"> <li>• Discuss the role of requirement analysis in AI integration projects</li> <li>• Review requirement elicitation techniques (interviews, observations)</li> <li>• Explore templates for functional and data requirements</li> <li>• Analyze a sample AI project requirement document</li> </ul>
	<b>Day 17</b> <ul style="list-style-type: none"> <li>• Functional &amp; non-functional requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Define functional requirements for AI systems (input, output, workflow)</li> <li>• Identify non-functional requirements (performance, scalability, ethics)</li> <li>• Use checklist to validate completeness</li> <li>• Group exercise: extract F/NF requirements from a case</li> </ul>
	<b>Day 18</b> <ul style="list-style-type: none"> <li>• Drafting a complete requirement spec for AI projects</li> </ul>	<ul style="list-style-type: none"> <li>• Choose a sample AI application (e.g., document classifier, object detector)</li> <li>• Identify stakeholders and user goals</li> <li>• Draft full spec: problem, scope, inputs, expected outputs</li> <li>• Peer review for completeness and clarity</li> </ul>
	<b>Day 19</b> <ul style="list-style-type: none"> <li>• Mapping requirements to Huawei AI tools (EI, ModelArts, HiLens)</li> </ul>	<ul style="list-style-type: none"> <li>• Map functional goals to Huawei Cloud services</li> <li>• Choose APIs/models aligned to use case</li> </ul>

		<ul style="list-style-type: none"> <li>Determine whether edge, cloud or hybrid is best</li> <li>Finalize tech stack and architecture sketch</li> </ul>
	<b>Day 20</b> <ul style="list-style-type: none"> <li>Assignment: Create a requirement document for a retail analytics system</li> </ul>	<ul style="list-style-type: none"> <li>Define use case (e.g., customer traffic analysis, shelf monitoring)</li> <li>Document all functional/NF requirements</li> <li>Map out service architecture using Huawei Cloud</li> <li>Submit and present summary to peer group</li> </ul>
Huawei Cloud ModelArts Introduction	<b>Day 21</b> <ul style="list-style-type: none"> <li>What is ModelArts: features and UI walk-through</li> </ul>	<ul style="list-style-type: none"> <li>Introduce ModelArts platform and its role in AI lifecycle</li> <li>Explore UI: datasets, training, deployment tabs</li> <li>Review service architecture and pricing tiers</li> <li>Navigate built-in tools: notebook, AutoML, pipelines</li> </ul>
	<b>Day 22</b> <ul style="list-style-type: none"> <li>Using pre-trained foundation models (vision, NLP)</li> </ul>	<ul style="list-style-type: none"> <li>Locate and explore ModelArts pre-trained models</li> <li>Run image classification and sentiment analysis demos</li> <li>Examine input/output formats and APIs</li> <li>Customize inputs and compare predictions</li> </ul>
	<b>Day 23</b> <ul style="list-style-type: none"> <li>Model training lifecycle: datasets, labeling, evaluation</li> </ul>	<ul style="list-style-type: none"> <li>Create dataset: upload or link from OBS</li> <li>Label dataset manually or via Auto-labeling</li> <li>Train a model with sample configuration</li> <li>Evaluate results and download metrics</li> </ul>
	<b>Day 24</b> <ul style="list-style-type: none"> <li>AutoML and experiment management in ModelArts</li> </ul>	<ul style="list-style-type: none"> <li>Use AutoML to build model from tabular/image data</li> <li>Compare multiple runs in experiment manager</li> <li>View loss curves and adjust settings</li> <li>Save and export best model</li> </ul>
	<b>Day 25</b> <ul style="list-style-type: none"> <li>Practical: Fine-tune a pre-trained sentiment model on a custom dataset</li> </ul>	<ul style="list-style-type: none"> <li>Select a text-based pre-trained NLP model</li> <li>Prepare and upload labeled custom reviews</li> <li>Fine-tune using ModelArts AutoML or training job</li> <li>Evaluate with accuracy/f1 and</li> </ul>

		download model
<b>HiLens Architecture &amp; Use Cases</b>	<b>Day 26</b> <ul style="list-style-type: none"> <li>HiLens overview: hardware + architecture</li> </ul>	<ul style="list-style-type: none"> <li>Explain HiLens edge AI use cases (retail, surveillance, logistics)</li> <li>Describe device architecture and deployment scenarios</li> <li>Review Huawei HiLens Kit specs and supported models</li> <li>Discuss edge vs cloud inference</li> </ul>
	<b>Day 27</b> <ul style="list-style-type: none"> <li>Features of HiLens: edge AI, video analytics, event handling</li> </ul>	<ul style="list-style-type: none"> <li>Walk through event-driven architecture in HiLens</li> <li>Explore supported video and image formats</li> <li>Understand stream processing and scheduling</li> <li>View real-time camera analytics demo</li> </ul>
	<b>Day 28</b> <ul style="list-style-type: none"> <li>HiLens SDK and deployment pipeline overview</li> </ul>	<ul style="list-style-type: none"> <li>Set up the HiLens development environment (SDK install, device link)</li> <li>Build and package a simple model</li> <li>Deploy model using HiLens pipeline builder</li> <li>Monitor status and debug device</li> </ul>
	<b>Day 29</b> <ul style="list-style-type: none"> <li>Practical: Deploy pre-built model to HiLens device</li> </ul>	<ul style="list-style-type: none"> <li>Choose a vision model (face detection, object tracking)</li> <li>Deploy to HiLens and stream test input</li> <li>Observe inference logs on device and cloud</li> <li>Record results and validate detection accuracy.</li> </ul>
	<b>Day 30</b> <ul style="list-style-type: none"> <li>Practical: Create simple event-triggered inference application</li> </ul>	<ul style="list-style-type: none"> <li>Define a business logic event (e.g., “detect person” = send alert)</li> <li>Write inference + action logic in SDK</li> <li>Upload, test, and validate real-time event output</li> <li>Document edge inference lifecycle</li> </ul>
<b>Deep Learning Concepts &amp; Frameworks</b>	<b>Day 31</b> <ul style="list-style-type: none"> <li>Neural networks: perceptrons, activation functions, layers</li> </ul>	<ul style="list-style-type: none"> <li>Introduce basic building blocks: neurons and layers</li> <li>Visualize forward propagation and loss</li> <li>Discuss activation functions (ReLU, Sigmoid, Softmax)</li> <li>Implement a basic neural net in pseudo-code</li> </ul>
	<b>Day 32</b> <ul style="list-style-type: none"> <li>CNNs and RNNs explained with visual demos</li> </ul>	<ul style="list-style-type: none"> <li>Explain CNN architecture for image tasks (filters, pooling)</li> <li>Walk through RNNs for sequential data</li> </ul>



		<ul style="list-style-type: none"> <li>Show visual demo of feature extraction and prediction</li> <li>Compare CNN vs RNN use cases.</li> </ul>
	<b>Day 33</b> <ul style="list-style-type: none"> <li>TensorFlow vs. PyTorch in ModelArts</li> </ul>	<ul style="list-style-type: none"> <li>Compare frameworks in terms of syntax, abstraction, flexibility</li> <li>Run a simple image classifier using both frameworks in notebooks</li> <li>Discuss when to choose one over the other</li> <li>Examine integration support in ModelArts</li> </ul>
	<b>Day 34</b> <ul style="list-style-type: none"> <li>Building a CNN with TensorFlow for image classification</li> </ul>	<ul style="list-style-type: none"> <li>Load and preprocess image dataset (e.g., Fashion MNIST)</li> <li>Define CNN architecture using Keras (TensorFlow)</li> <li>Train and validate model</li> <li>Plot metrics and review misclassifications</li> </ul>
	<b>Day 35</b> <ul style="list-style-type: none"> <li>Practical: Train &amp; test food classification model using ExeML</li> </ul>	<ul style="list-style-type: none"> <li>Upload food image dataset to ModelArts</li> <li>Use ExeML to auto-train image classifier</li> <li>Monitor training progress and view performance graphs</li> <li>Test on new images and download inference results</li> </ul>
<b>Model Deployment &amp; Integration</b>	<b>Day 36</b> <ul style="list-style-type: none"> <li>ModelArts deployment methods: real-time vs. batch</li> </ul>	<ul style="list-style-type: none"> <li>Explain difference between real-time and batch inference</li> <li>Walk through deployment types in ModelArts</li> <li>Select appropriate deployment for use case</li> <li>Explore latency vs throughput trade-offs</li> </ul>
	<b>Day 37</b> <ul style="list-style-type: none"> <li>API endpoints, SDKs, and integration best practices</li> </ul>	<ul style="list-style-type: none"> <li>Expose deployed model as API endpoint</li> <li>Use Python SDK to authenticate and send input</li> <li>Parse output and handle errors</li> <li>Review common security and versioning practices</li> </ul>
	<b>Day 38</b> <ul style="list-style-type: none"> <li>Practical: Deploy and call a model endpoint from a Python script</li> </ul>	<ul style="list-style-type: none"> <li>Write a Python client to send request to deployed API</li> <li>Format input based on model type (text/image)</li> <li>Print and log model predictions</li> <li>Test with multiple inputs and handle exceptions</li> </ul>
	<b>Day 39</b> <ul style="list-style-type: none"> <li>Case Study: Goods Recognition in</li> </ul>	<ul style="list-style-type: none"> <li>Review case architecture: camera + HiLens + ModelArts API</li> </ul>

	shopping malls	<ul style="list-style-type: none"> <li>Analyze model logic for object detection &amp; tagging</li> <li>Walk through edge-cloud inference chain</li> <li>Explore data flow and latency implications.</li> </ul>
	<b>Day 40</b> <ul style="list-style-type: none"> <li>Practical: Build and test retail goods recognition app</li> </ul>	<ul style="list-style-type: none"> <li>Simulate a product recognition app using image uploads</li> <li>Trigger inference through API</li> <li>Display recognized goods and confidence scores</li> <li>Store results for analytics (optional DB or CSV)</li> </ul>
<b>Testing &amp; Optimization for AI Apps</b>	<b>Day 41</b> <ul style="list-style-type: none"> <li>Testing methods: unit, integration, system testing in AI</li> </ul>	<ul style="list-style-type: none"> <li>Introduce testing layers in AI pipelines</li> <li>Design unit tests for pre/postprocessing functions</li> <li>Discuss integration tests across API/data flow</li> <li>Identify system testing checkpoints for full pipeline</li> </ul>
	<b>Day 42</b> <ul style="list-style-type: none"> <li>Role of testing in model iteration and optimization</li> </ul>	<ul style="list-style-type: none"> <li>Define model performance testing (accuracy, latency)</li> <li>Explore dataset versioning and re-training triggers</li> <li>Create A/B test plans for two models</li> <li>Discuss how testing informs tuning</li> </ul>
	<b>Day 43</b> <ul style="list-style-type: none"> <li>Performance benchmarking and error analysis</li> </ul>	<ul style="list-style-type: none"> <li>Collect prediction logs and latency data</li> <li>Use confusion matrix for classification error analysis</li> <li>Identify false positives/negatives and root causes</li> <li>Adjust threshold or retrain with targeted samples</li> </ul>
	<b>Day 44</b> <b>IoT Device Integration with Edge AI</b> <ul style="list-style-type: none"> <li>Practical: Write test cases for OCR and classification APIs</li> </ul>	<ul style="list-style-type: none"> <li>Define test inputs and expected results for OCR</li> <li>Run classification API on noisy data</li> <li>Validate output format and correctness</li> <li>Document edge cases and inconsistencies</li> </ul>
	<b>Day 45</b> <ul style="list-style-type: none"> <li>Practical: Analyze test logs and suggest model improvements</li> </ul>	<ul style="list-style-type: none"> <li>Review logs for latency spikes or failed calls</li> <li>Correlate performance issues with input variations</li> <li>Recommend dataset refinements</li> </ul>

		or model architecture changes <ul style="list-style-type: none"> <li>• Prepare a testing summary report</li> </ul>
<b>MLOps &amp; CI/CD on Huawei Cloud</b>	<b>Day 46</b> <ul style="list-style-type: none"> <li>• Introduction to MLOps &amp; pipeline orchestration</li> </ul>	<ul style="list-style-type: none"> <li>• Define MLOps lifecycle stages: data, model, code, deployment</li> <li>• Identify tools used in Huawei Cloud for orchestration</li> <li>• Map traditional DevOps to AI workflows</li> <li>• Review examples of automated pipelines</li> </ul>
	<b>Day 47</b> <ul style="list-style-type: none"> <li>• CI/CD integration using ModelArts + OBS + FunctionGraph</li> </ul>	<ul style="list-style-type: none"> <li>• Configure OBS bucket for model/data storage</li> <li>• Set up ModelArts pipeline triggered by data arrival</li> <li>• Use FunctionGraph to automate post-inference task (e.g., notification)</li> <li>• Run CI/CD loop and monitor</li> </ul>
	<b>Day 48</b> <ul style="list-style-type: none"> <li>• Monitoring &amp; model drift detection</li> </ul>	<ul style="list-style-type: none"> <li>• Explain model drift (concept and data)</li> <li>• Configure CloudEye for monitoring inference performance</li> <li>• Set thresholds for retraining based on accuracy drop</li> <li>• Log input distributions for comparison</li> </ul>
	<b>Day 49</b> <ul style="list-style-type: none"> <li>• Practical: Build a simple data drift-triggered retraining pipeline</li> </ul>	<ul style="list-style-type: none"> <li>• Simulate drift with changed dataset</li> <li>• Detect drop in model performance</li> <li>• Trigger training pipeline via event</li> <li>• Deploy updated model and compare before/after metrics</li> </ul>
	<b>Day 50</b> <ul style="list-style-type: none"> <li>• Group discussion: MLOps challenges and solutions</li> </ul>	<ul style="list-style-type: none"> <li>• Present real-world MLOps pain points (e.g., data lag, brittle models)</li> <li>• Discuss reproducibility and traceability</li> <li>• Compare open-source vs Huawei pipeline tools</li> <li>• Share group recommendations and summaries</li> </ul>
<b>Exam Prep &amp; Certification Focus</b>	<b>Day 51</b> <ul style="list-style-type: none"> <li>• Full-length mock exam 1 + Review</li> </ul>	<ul style="list-style-type: none"> <li>• Attempt 60-minute mock exam simulating real conditions</li> <li>• Score and record performance by domain</li> <li>• Review difficult questions in group</li> <li>• Trainer explains key mistakes and strategies</li> </ul>

	<b>Day 52</b> <ul style="list-style-type: none"> <li>Domain-wise question practice: APIs, ModelArts, HiLens</li> </ul>	<ul style="list-style-type: none"> <li>Solve 5–10 questions per domain (with reasoning)</li> <li>Group discussion: compare thought processes</li> <li>Focus on OCR, ModelArts UI, API endpoints</li> <li>Recap of certification syllabus</li> </ul>
	<b>Day 53</b> <ul style="list-style-type: none"> <li>Mock exam 2 + Discussion</li> </ul>	<ul style="list-style-type: none"> <li>Take a second 60-min mock with new questions</li> <li>Score and self-assess improvement vs Day 51</li> <li>Debrief: what to revise in remaining time</li> <li>Share individual strategy plans</li> </ul>
	<b>Day 54</b> <ul style="list-style-type: none"> <li>Final tips, time management, and last-minute revision</li> </ul>	<ul style="list-style-type: none"> <li>Prioritize key topics (e.g., deployment, API integration)</li> <li>Practice time-boxing and flagging strategy</li> <li>Use flashcards / cheat sheet for memorization</li> <li>Q&amp;A session for final clarifications</li> </ul>
	<b>Day 55</b> <ul style="list-style-type: none"> <li>Group Q&amp;A session, problem-solving clinic</li> </ul>	<ul style="list-style-type: none"> <li>Raise personal doubts from any week/module</li> <li>Solve peer-posted practical or theoretical cases</li> <li>Trainer reviews mock stats and offers guidance</li> <li>Calm nerves: talk exam-day logistics and mindset</li> </ul>
<b>Capstone Project Development &amp; Review</b>	<b>Day 56</b> <ul style="list-style-type: none"> <li>Project kickoff: choose a domain (e.g., chatbot, smart retail, inspection)</li> </ul>	<ul style="list-style-type: none"> <li>Review sample project ideas and constraints</li> <li>Define objectives, data needs, and success criteria</li> <li>Allocate teams or solo roles</li> <li>Draft system architecture</li> </ul>
	<b>Day 57</b> <ul style="list-style-type: none"> <li>Build project: Data ingestion, training, API deployment</li> </ul>	<ul style="list-style-type: none"> <li>Collect or upload training dataset</li> <li>Train initial model using ModelArts or ExeML</li> <li>Deploy as API or HiLens stream</li> <li>Begin writing inference client app</li> </ul>
	<b>Day 58</b> <ul style="list-style-type: none"> <li>Build project: Testing and presentation draft</li> </ul>	<ul style="list-style-type: none"> <li>Write unit and integration test cases</li> <li>Evaluate performance metrics and make improvements</li> <li>Finalize visual presentation: code + results</li> <li>Conduct dry-run with peer feedback</li> </ul>

	<b>Day 59</b> <ul style="list-style-type: none"> <li>• Capstone project presentations + code demo</li> </ul>	<ul style="list-style-type: none"> <li>• Present architecture, training pipeline, results</li> <li>• Demo model in action (API call or HiLens test)</li> <li>• Answer Q&amp;A from peers and instructors</li> <li>• Submit all artifacts (slides, code, logs)</li> </ul>
	<b>Day 60</b> <ul style="list-style-type: none"> <li>• Feedback, certification registration support, closing &amp; celebration</li> </ul>	<ul style="list-style-type: none"> <li>• Receive feedback and scores from instructors</li> <li>• Share success stories and lessons learned</li> <li>• Complete official exam registration (if not done)</li> <li>• Certificate distribution + closing remarks</li> </ul>

## LIST OF MACHINERY / EQUIPMENT

For the Class of 25 - 35 Students (3-Month Course)

S. No	Name of Items	Quantity/Unit
1.	Intel i5/i7, 8 GB Ram, 250 GB ssd	35
2.	High-Speed Internet Min 10 MB CIR	1
3.	Monitors (24-inch FHD/IPS)	2
4.	Backup Power Supply (UPS)	1
5.	Projector / Smart Display	1
6.	Microphone & Speaker System	1

## LIST OF CONSUMABLE MATERIAL

For the Class of 30 Students (3-Month Course)

S. No	Name of Items	Unit
1.	Writing Notebooks	30
2.	Pens (Blue/Black)	60
3.	Markers (Whiteboard)	10
4.	Whiteboard Dusters	2
5.	Printing Paper (A4)	5 Rims
6.	File Folders	30