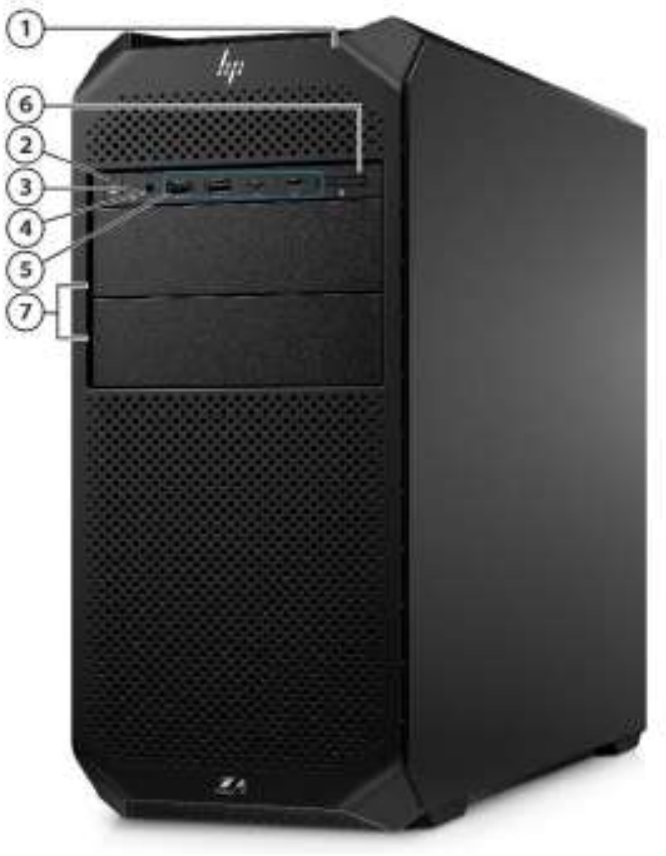
SIGMA

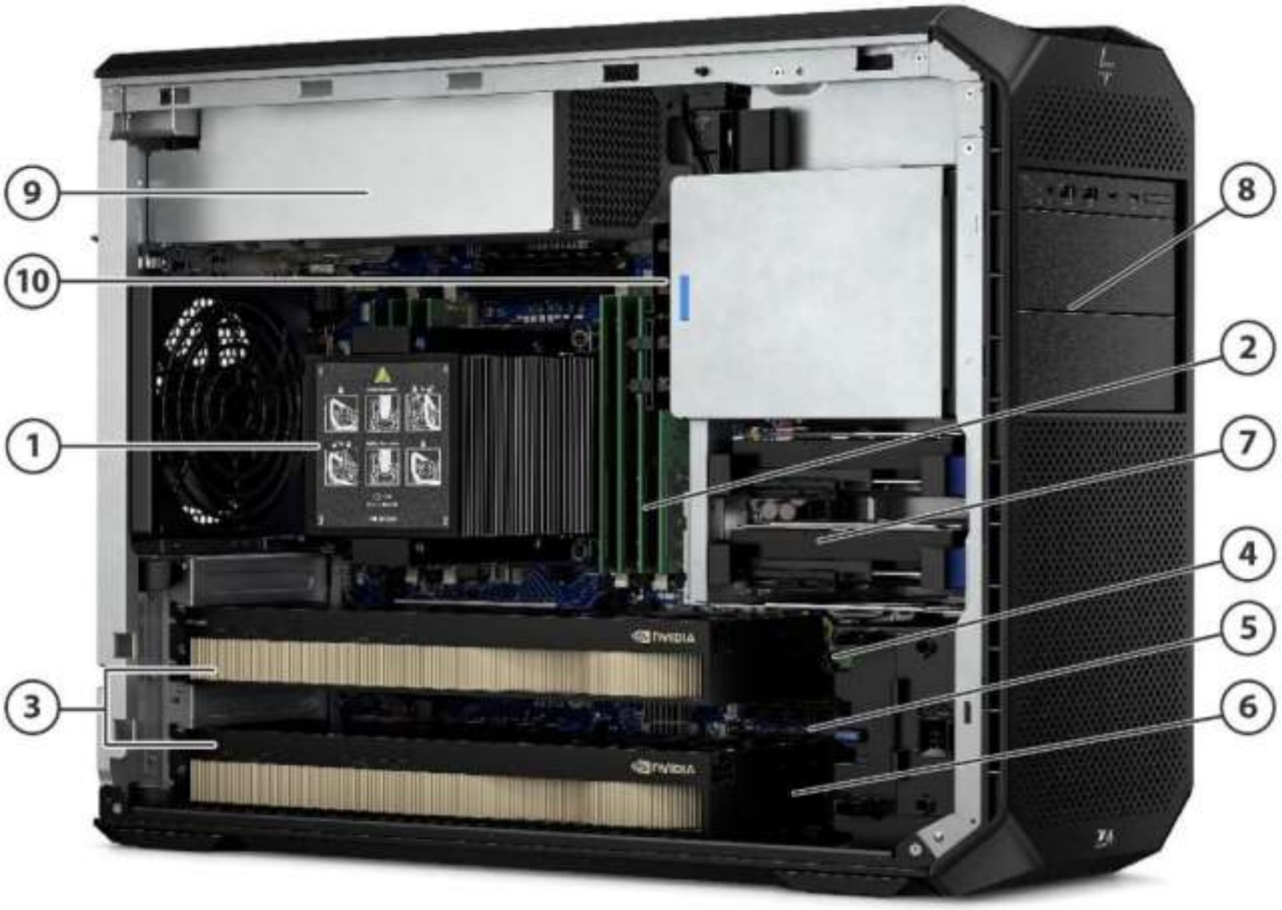
AI DESKTOP WORK STATION

MODEL-AI-DESKTOP100

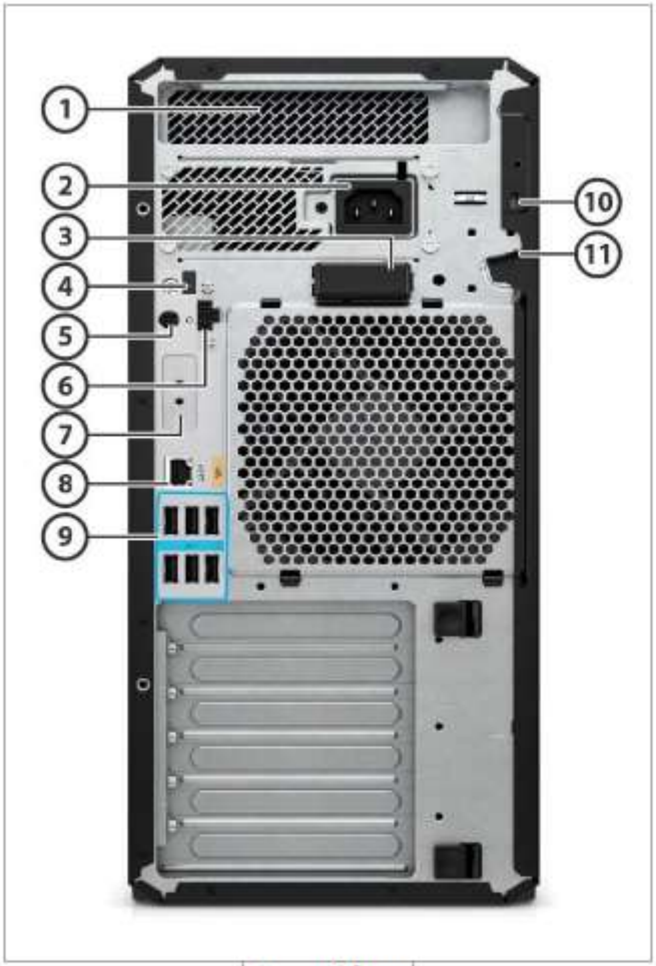
SPECIFICATIONS



1. Integrated Front Handle
2. Power Button
3. HDD Activity LED
4. Headphone / Microphone Combo
5. USB Type -C 20 Gbps, USB 3.0, USB Type-A
6. SD Card Reader
7. 2 x External 5.25" Bay



1. Intel® Xeon® Processor
2. 4 DIMM slots for DDR5 ECC Memory
3. PCIe x16 Port
4. PCIe x4 Gen4 configurable with M.2 SSD
5. Sata Ports
6. Internal USB Ports
7. 2 Internal 3.5" Bays
8. 2 External 5.25" Bays
9. 850 W SMPS Slot



Rear View

1. Integrated Rear Handle
2. Power Connector
3. Provision for External Antenna
4. Rear Power Button
5. Audio In/Out
6. Manageability Port (Optional)
7. Flex I/O Module (Optional)
8. RJ-45 Integrated LAN Port
9. 6 SuperSpeed USB Type-A
10. Kensington Lock Slot
11. Padlock loop

This Desktop Work Station Computer has been designed with a view to provide practical and experimental knowledge of Artificial Intelligence (AI) and Machine Learning (ML) technology.

SPECIFICATIONS

A. Hardware:

1.	CPU	Intel Xeon 6 Core – E2336
2.	Mother Board	Gigabyte MX33
3.	Memory	4 x 16GB DDR4 ECC RAM = Total 64 GB
4.	Hard Disk	2 TB Sata HDD – 1 No and 240 GB SSD – 1 No.
5.	GPU	NVIDIA RTX A4000 – 16 GB – 1 No
6.	SMPS	850W SMPS Power supply
7.	Cabinet	Tower Mount Cabinet chassis
8.	Mouse	Logitech USB Mouse
9.	Keyboard	TVSE Gold USB Keyboard
10.	Monitor	15 Inch LED with HDMI Input

B. Software:

1. Dual Boot Operating system
2. Operating System : Ubuntu OS - 22.04 and Windows 11

C. Pre-installed Libraries, Utilities, Tools and SDKs

A. Ubuntu OS

1.	Essentials Utilities	CUDA, cuDNN, TensorRT
2.	Machine Learning Libraries	Vowpal Wabbit, XGBoost, Numpy, Scikit, Pandas and other Pylibs
3.	Deep Learning Libraries	NVidia DIGITS, Tensor Flow, Caffe, Caffe2, PyTorch, Torch, Theano CNTK, Dockers, Containers
4.	Dataset	ImageNet, CIFAR-10, KITTI, COCO, Kaggle
5.	Images and Video Processing	OpenCv, OpenGL, Vulkan, CUDA, VisionWorks
6.	Neural Networks	GoogleNet, MobileNet, ResNet, Inception

B. Wndows-11 OS – Eco System – Edge Computing Platform

1. Windows 11 Operating system based IoT ecosystem
2. Arm Cortex
3. Linux
4. Cortex-M4 IDE
5. Arduino IDE
6. Eclipse IDE
7. RTos
8. Mbed
9. OSGi Framework
10. OpenSource IOT
11. GNU Toolchain
12. Flashing Tools
13. Communication Tools

D. Training Programs

1. Exclusive Training on Deep Learning
2. Training on AI-DL Machine Hardware Setup
3. Training on Docker & Container based environment installed
4. Operation of Software Tools / Utilities & Environments enabling multiuser usage environment

E. Admin Training:

1. Login to AI-DL Machine as Administrator
2. Managing User accounts,
3. User Creation and Deletion
4. Hardware Parameter Visualization
5. Managing Hardware Resources
6. Training on Docker and Container
7. Checking available Docker Images
8. Adding and Managing various Libraries and Software versions in specific Docker Images
9. Backup and Restore Docker Images
10. Backup and Restore user Data to and from user space to and from server database
11. Script based management for ease of operation

F. User Training

1. Login to AI-DL Machine as Administrator
2. Training on Docker and Container
3. Checking available Docker Images
4. How to User Docker Images
5. Check available Libraries and Software
6. Running simple example using Digits
7. Running simple example using Jupyter Notebook
8. How to run your own example on AI-DL Machine
9. Data Management
10. Transferring DATA to and from AI-DL Computing Machine to and from user node machine
11. Manage storage space

G. Accessories

1.	All Cables and Adaptors	
2.	Pen Drive	: 16 GB with All Codes and Soft copy of Manual
3.	E-Books for AI, ML, DL Subject	: 100 Nos. in PDF Format
4.	Mp4 Video for AI, ML, DL Subject	: 100 Nos
5.	Online Cloud/Server Services	: For 2 Years on Cloud Server
6.	Live Training at College	: For 2 Days for 4 Hours per Day
7.	After Sale Training support	: By Online Zoom Meeting or By Whatsapp Video Call

1. ARTIFICIAL INTELLIGENCE – AI - EXPERIMENTS

1. AI Operating Systems

1. Ubuntu 22.04.LTS
2. Windows 11 Pro
3. NVIDIA JetPack having Board support package - BSP

2. AI Languages -

1. Python
2. C, C++
3. R
4. Visual Basic Studio
5. MicroPython
6. Soda OS
7. Jetson SDK

3. Libraries

1.	Python	3.10.12
2.	cuda	12.2.142
3.	cuDNN	8.94.25
4.	cython	3.7.0
5.	gdown	4.7.1
6.	Git	12.34.1
7.	h5py	3.10.0
8.	imutils	0.5.4
9.	jetson-stats	4.2.4
10.	matplotlib	3.5.1
11.	numpy	1.26.3
12.	pandas	2.1.4
13.	pillow	9.0.1
14.	pip	22.0.2
15.	protobuf	4.23.4
16.	psutil	5.9.0
17.	pycuda	2021.1
18.	pyserial	3.5
19.	pyTools	2023.1.1
20.	PyYAML	5.4.1

21.	Scikit-learn	0.23.2
22.	Scipy	1.8.0
23.	Seaborn	0.11.2
24.	Setuptools	59.6
25.	Tensorrt	8.6.1
26.	Tqdm	4.66.1
27.	Wheel	0.37.1
28.	Cmake	3.22.1
29.	Jupyter Notebook	1.0.0

4. Frameworks

1.	CUDA	12.2.142	
2.	cuDNN	8.94.25	
3.	TensorRT	8.6.1	
4.	Torch and PyTorch	2.1.2	
5.	TorchVision	0.16.2	
6.	TensorFlow	2.15.0	- To make AI Frame work
7.	Keras	2.15.0	- For High Performance Numerical Computation
8.	Caffe		
9.	Caffe2		
10.	DL4J		
11.	NVidia Vision Works		
12.	NVIDIA DIGITS		
13.	Vowpal Wabbit		
14.	Vulkan		
15.	Xgboost		
16.	Theano		
17.	CNTK		
18.	Kubernet		
19.	Dockers and Containers		

5. Dashboards

1. Jupyter Notebook
2. Google Colab
3. Anaconda
4. Conda
5. GitHub

6. AI Mathematics

1. Linear Algebra – Linear Equations, Matrixs, Vectors
2. Calculus – Differentiation, Integration, Gradient Descent,
3. Statistics – Population, Parameter, Sample, Variable, Probability

7. AI Applications

1.	Computer Vision	4.8.1.78	For real-time Image Processing
2.	YOLO5	5.0	For Object detection
3.	OpenCV – cv2	4.9.0	For Image processing
4.	Open GL		
5.	Vulkan		
6.	NVidia Vision Works		
7.	CUDA		

8. Cloud Services – (To open paid account by Customer with AWS and Google)

1. Google Personal Cloud - GPC Services
2. Google Colab Account
3. Amazon Web services – AWS
4. MS Azure Web services
5. Nvidia GCP
6. CHATGPT Services
7. Dall-E Services

9. AI Datasets

1. CIFAR-10
2. ImageNet
3. KITTI
4. Kaggele
5. Alexnet
6. DarkNet
7. GoogelNet
8. Hydranets
9. MobielNet V2
10. MobileNetSS
11. MxNet
12. PoseNet
13. SqueezeNet
14. U-Net

15. VGGNet
16. ResNet
17. COCO

10. AI Experiments

1. To Run and understand Basic Python Program codes
2. To connect USB camera and take Audio, Video and Image pictures
3. To connect CSI camera and take Audio, Video and Image pictures
4. To demonstrate Jupyter Notebook procedures and run .py and .ipynb files
5. To detect object images using OpenCV – cv2
6. To use Computer Vision to track Person Faces, Distance, Objects etc.
7. To use YOLO5 for image and Video Processing images and videos
8. To run TensorFlow and Keras for mathematical calculation of Matrixes and Tensors
9. To run different .ipynb files for different Machine and Deep leading programs

11. OpenCV and Computer Vision Experiments

1. Face Detection and Tracking
2. Face Recognition
3. Emotion Recognition
4. Gesture Recognition
5. Smile Detection
6. Vehicle Detection
7. Object Detection using Yolo algorithm
8. Drowsiness Detection
9. License Plate Detection
10. Fingerprint Recognition
11. Text identification
12. Traffic Sign Recognition
13. Motion Detection
14. Character Recognition
15. Edge Detection through Image processing
16. Handwritten Digit Classification using CNN
17. Leaf Disease Detection and Classification
18. Pattern Recognition
19. Fire Detection
20. Weather Forecasting

12. AI Applications in Real Life

1. Natural Language Processing – NLP
2. Internet of Things – IOT
3. Preventive Maintenance
4. Cyber Security
5. Agriculture and Food Industry
6. Remote Healthcare Monitoring and Telemedicine
7. Environment Monitoring and Forecast
8. Warehouse and Logistics Monitoring
9. Retail Analysis
10. Intelligent Traffic Management
11. Energy Monitoring and Control
12. Home and Building Automation
13. GPU computing
14. Multimedia Processing

13. Database Servers

1. InfluxDB
2. MapDB
3. MySQL

14. Types of AI

1. Generative AI - GEN AI
2. Edge AI
3. Conversational AI
4. Open AI
5. Applied AI
6. CHATGPT AI
7. Google Bard - CHATGPT
8. Artificial General Intelligence
9. Large Language Model – LLM
10. GoogleAI

2. MACHINE LEARNING - ML – EXPERIMENTS

1. Supervised Learning

1. Linear Regression
2. Logistic Regression
3. Polynomial Regression
4. Gradient Descent
5. Decision Trees
6. Random Forest
7. Bagging & Boosting
8. K Nearest Neighbors – KNN
9. Bayesian Linear Regression
10. Non-Linear Regression
11. Support Vector Machine
12. Newton's Method
13. MLE (Maximum Likelihood Estimation)
14. MAP (Maximum A Posteriori)
15. PCA (Principal Component Analysis)
16. L1 Regularization (Lasso Regression)
17. L2 Regularization (Ridge Regression)

2. Unsupervised Learning

1. K-Means
2. Hierarchal Clustering

3. Reinforcement Learning

4. Ensemble Learning

5. Other Experiments

1. To understand theory of Gaussian Mixture Model – GMM
2. To understand theory of Support Vector Machine - SMM
3. To understand theory of MLOps – Machine Learning Operations
4. To understand theory of DevOps - Developments and Operations
5. To understand theory of PCA - Principal Component Analysis
6. To understand theory of Cost Function

7. To understand theory of Text Classification Using Naive
8. To understand theory of Polynomial Regression Algorithm
9. To understand theory of Back propagation and Gradient Descent
10. To understand theory of Filters used to detect Spam Emails
11. To understand theory of Entropy In Decision Tree Intuition
12. To understand theory of Gini Impurity Intuition In Depth In Decision Tree
13. To understand theory of Ensemble - What is Bagging (Bootstrap Aggregation)
14. To understand theory of DBSCAN Clustering
15. To understand theory of Silhouette Clustering
16. To understand theory of What is Cross Validation and its types
17. To understand theory of Bayes' Theorem for Conditional Probability
18. To understand theory of Xgboost Regression
19. To Deploy ML Models using PyWebIO and Flask in Heroku
20. To understand theory of R Squared Theory
21. To understand theory of Euclidean Distance
22. To understand theory of Overfitting And Underfitting Machine Learning
23. To understand theory of Gaussian Mixture Model
24. To understand theory of Machine Learning Life Cycle
25. To understand theory of Supercharging Decision Making with Bayes
26. To understand theory of Multiple Linear Regression
27. To understand theory of Q-Learning Agent Analysis in Reinforcement Learning
28. To understand theory of Deep Reinforcement Learning (DQN)

6. Machine Learning Projects

1. Building a Sales Prediction using Machine Learning
2. Customer Segmentation using Machine Learning
3. Mastering Sentiment Analysis with Machine Learning and Flask
4. Credit Card Fraud Detection using Machine Learning
5. Credit Card Risk Assessment using Machine Learning
6. Hate Speech Detection Using Machine Learning
7. Fake News Detection Using Machine Learning
8. Predicting Heart Disease using Machine Learning
9. Diabetes Prediction using Machine Learning
10. Parkinson’s Disease Detection using Machine Learning

11. Faringham Disease Prediction Using Machine Learning
12. DNA Sequencing Classifier using Machine Learning
13. Breast Cancer Classification with Machine Learning
14. Email Spam Detection
15. Autocorrect Spell Checking using Machine Learning
16. Building Grammar and Spell Checker using Machine Learning
17. Crop Recommendation System using Machine Learning
18. How Netflix Uses Machine Learning to Show movie Prediction
19. Movie Recommender System Using Machine Learning
20. How Does YouTube Recommend Videos
21. Building a Music Recommendation Engine
22. Amazon products recommendations system using Machine Learning

3. DEEP LEARNING - DL – EXPERIMENTS

1. Deep Learning Experiments

1. To understand theory of Artificial Neural Networks - ANN
2. To understand theory of Convolutional Neural Network - CNN
3. To understand theory of Recurrent Neural Network - RNN
4. To understand theory of Generative Adversarial Network – GAN
5. To understand theory of Graph Neural Network - GNN
6. To understand theory of StyleGAN Network – SGNN
7. To understand theory of CycleGAN - CGNN
8. To understand theory of Deep Convolutional Generative Adversarial Network – DCGAN
9. To understand theory of GauGAN
10. To understand theory of Unconventional Neural Networks – UNN
11. To understand theory of Multiple Neural Networks – MNN
12. To understand theory of Mask R-CNN
13. To understand theory of Faster R-CNN
14. To understand theory 3D Convolutional Neural Network
15. To understand theory of Think Neural Network – THNN
16. Fully Convolutional Networks (FCN)
17. To understand theory of Movidius NCS - Neural Computer Stick
18. To understand theory of Single Shot Detector Code for SSD Model
19. To understand Complex Mathematics used for Deep Learning
20. To understand theory of LeNet Architecture
21. To understand theory of ResNet Using Keras Residual Network
22. To understand Deep Learning in Medical Science
23. Predicting Lungs Disease using Deep Learning
24. To use 3D Convolutional Neural Network for Lung Cancer Detection
25. Malaria Disease Detection using Deep Learning
26. To understand Long Short Term Memory – LSTM - with Example
27. Stock Price Prediction and Forecasting using Stacked LSTM - Deep Learning
28. Monte Carlo Dropout Layers In Deep Learning
29. To understand - What Is Transfer Learning in Deep Learning
30. To demonstrate Neural Networks
31. To demonstrate Convolutional Neural Networks

32. To demonstrate Plant leaf disease detection using Mask R-CNN Image Segmentation
33. To demonstrate Deep Learning Applications in Real Life
34. To demonstrate Backpropagation and Gradient Descent In Neural Networks
35. To demonstrate Various Weight Initialization Techniques in Neural Network
36. Training Neural Networks on GPU vs. CPU Performance Test
37. How to choose number of hidden layers and nodes in Neural Network
38. Encoder And Decoder- Neural Machine Learning Language Translation With Keras
39. Instance Segmentation Using Mask R-CNN on Custom Dataset
40. Instance Segmentation Web Application Using Mask R-CNN and Flask
41. Video Classification with a CNN-RNN Architecture for Human Activity Recognition
42. Next Word Prediction using RNN like WhatsApp application
43. Language Translator using seq2seq Model with RNN

2. Deep Learning Applications

1. Image Classification with DIGITS
2. Object Detection with DIGITS
3. Object Detection over KITTI dataset with DIGITS
4. Semantic Segmentation using DIGITS
5. Medical Image Segmentation using DIGITS
6. Signal Processing using DIGITS
7. Train a Generative Adversarial Network using DIGITS
8. Training an image auto encoder with DIGITS
9. Binary Segmentation using DIGITS
10. Linear Classification with Tensor Flow
11. Image Classification using Tensor Flow
12. Demonstration of remote inference of Deep Learning model using Embedded GPU board

4. NATURAL LANGUAGE PROCESSING – NLP – EXPERIMENTS

1. NLP Applications

1. To understand theory of audio processing
2. To understand theory of AI Voice Assistance
3. To understand theory of AI Chatbot
4. To understand theory of Audio Fingerprinting
5. To understand theory of Music Recommendation
6. To understand theory of Speech Recognition
7. To understand theory of Sentiment Analysis
8. To understand theory of Dialog Flow – Chatbot
9. To understand theory of Text Classification
10. To understand theory of Language Translation using NLP
11. To understand theory of Named Entity Recognition - NER
12. To understand theory of Minutes LangChain
13. To implement Word Embedding using Keras
14. To implement Travel Agency Chat Bot using Flask
15. To understand Rasa Chatbot with Database and store data of Chatbot in Database
16. To implement Chat Bot using Weather API
17. To demonstrate Basic Chatbot Using Rasa NLU and Rasa Core
18. To demonstrate AI Voice Assistance using NLP
19. To demonstrate AI Chatbot using NLP
20. To build a Chatbot with GUI in Python with Tkinter
21. To demonstrate Speech Recognition using NLP
22. To demonstrate Text Classification using NLP
23. To demonstrate Chatbot with Mic input Speaker output using Python, Jarvis, and DialoGPT
24. To build an AI Voice Assisted using ChatGPT
25. To record and play Audio signal using PyAudio
26. To demonstrate Text to Speech (TTS) Conversion
27. To understand GPT-3, GPT-NeoX and GPT-NeoX-20B models
28. To understand Natural Language Processing Tokenization
29. To understand Natural Language Processing - Stemming And Lemmatization Intuition
30. To understand Natural Language Processing TF-IDF Intuition - Text Preprocessing
31. To build Resume Analyser Application using NLP and Python
32. To build Keywords Extraction app with Python
33. To build ruled based Chabot in Python from Scratch

5. ADVANCE IMAGE AND VIDEO PROCESSING EXPERIMENTS USING GPU

1. Computer Vision – OpenCV Experiments

1. Write a program to display Hello World.
2. Write a program to Read image apply sobel filter and display output image.
3. Write a program to Read image apply sobel filter and display output image.
4. Write a program to Read Video file and display video.
5. Write a program to Read Video file apply sobel filter and write xvid video file.
6. Write a program to capture image from USB Webcam, apply sobel filter to it and write to image file
7. Write a program to capture video from USB Webcam, apply sobel filter to it and write to xvid video file.
8. Write a program to perform basic operation like resize over an image.
9. Write a program to perform Simple Canny Edge filter over an image.
10. Write a program to perform Canny Edge filter using blur technique to get desired result.
11. Write a program to perform simple feature detection using OrbFeatureDetector over a video.
12. Write a program to perform optical flow over feature detection to track features and show the tracking over a video.
13. Write a program to perform object detection by comparing unique points of an object to a video and find the object.
14. Write a program to perform object detection and match unique descriptors of object with video and draw lines to show match.
15. Write a program to perform object detection using matching with desired object and put a box around if the object is near to what is described.
16. Write a program to perform Face detection using Cascade Classifier.
17. Write a program to perform Face detection using Cascade Classifier with Histogram.
18. Write a program to perform Face detection using alternate Cascade Classifier profile.
19. Write a program to perform Background and foreground segmentation using CPU.
20. Write a program to perform Background and foreground segmentation using GPU.
21. Write a program to perform Laplace point edge detection using USB Webcam capture.
22. Write a program to perform Houghlines detection over an art image using both CPU and GPU.
23. Write a program to perform grabcut segmentation over selected section.

2. Computer Graphics – OpenGL Experiments

1. Write a program to demonstrate generation of large number of slightly varying objects with bindless rendering.
2. Write a program to demonstrate blooming effect on rendered surfaces making it glow.
3. Write a program to demonstrate access to GL textures using both reading and writing to image.
4. Write a program to demonstrate particle expansion by accessing vertex shaders in parallel.
5. Write a program to demonstrate water simulation by using compute shaders.
6. Write a program to demonstrate use of vertex shaders to animate particles and write back result into vertex buffer.
7. Write a program to demonstrate use of high performance and quality approximation of anti-aliasing.
8. Write a program to demonstrate High Dynamic Range (HDR) imaging.
9. Write a program to implement instancing to tessellate objects in real time.
10. Write a program to implement instancing to accelerate drawing of simillar objects
11. Write a program to demonstrate multi-pass filtering for motion blur of fast moving objects.
12. Write a program to demonstrate motion blur using 2D multi-pass filter.
13. Write a program to demonstrate large number of drawcalls overhead using openGL extension.
14. Write a program to implement openGLPSI (Pixel Shader Interlock) feature to blend decals.
15. Write a program to implement Path rendering extension to draw 2D line art.
16. Write a program to use path rendering and animate live cursive writing.
17. Write a program to implement HarfBuzz text shaping engine library for strings of unicode.
18. Write a program to implement path rendering to draw text like spokes in a wheel with 3D effect.
19. Write a program to implement conventional 3D graphic of Tiger using path rendering.
20. Write a program to implement path rendering to wrap an artwork of Tiger with multiple paths and animate.
21. Write a program to implement optimization techniques to rendering process to improve app level CPU GPU timings.
22. Write a program to simulate a cloud of particles and render its shadow on model or floor object.
23. Write a program to implement skinned meshes over bones in vertex shaders for smooth deformation.
24. Write a program to demonstrate two methods of simulating soft shadows.
25. Write a program to implement terrain engine by using hardware tessellation.
26. Write a program to implement Terrain using Texture Array for high performance.
27. Write a program to implement rendering of OIT (Order Independent Transparency) using weighted blending.

3. Computer Graphics – Vulkan Experiments

1. Write a program to render colored triangle on the screen.
2. Write a program to demonstrate the use of pipeline state objects (pso) in one single renderpass.
3. Write a program to demonstrate the use of descriptor sets for passing data to shader stages.
4. Write a program to demonstrate the use of Dynamic uniform buffers for rendering multiple objects with multiple matrices stored in a single uniform buffer object.
5. Write a program to demonstrate the use of (push constants) small shader block accessed outside of uniforms for fast updates.
6. Write a program to demonstrate the use of Shader specialization constants to create multiple pipelines with different lighting paths from a single "uber" shader.
7. Write a program to demonstrate texture loading including MIP maps.
8. Write a program to demonstrate the use of cube map textures.
9. Write a program to demonstrate the use of texture arrays to display 2D textures.
10. Write a program to generate a 3D texture.
11. Write a program to load Model and texture maps.
12. Write a program to demonstrate the use of sub pass to implements a deferred rendering setup with a forward transparency pass
13. Write a program to demonstrate the use of offsceen rendering to render mirror surface from the original image.
14. Write a program to implement a simple CPU based particle system.
15. Write a program to demonstrate the use of stencil buffer and its compare functionality for rendering a 3D model with dynamic outlines.
16. Write a program to aenders a scene made of multiple parts with different materials and textures (Scene rendering).
17. Write a program to implements multisample anti-aliasing (MSAA) using a renderpass with multisampled attachments and resolve attachments that get resolved into the visible frame buffer.
18. Write a program to implements a high dynamic range rendering pipeline using 16/32 bit floating point precision for all internal formats, textures and calculations, including a bloom pass, manual exposure and tone mapping.
19. Write a program to rendering shadows for a directional light source. (Shadow mapping)
20. Write a program to implement projective cascaded shadow mapping for directional light sources (Cascaded shadow mapping)
21. Write a program to implement omni directional shadows using a dynamic cube map.
22. Write a program to demonstrate how to generate a complete texture mip-chain using texture mapping at runtime instead of loading offline generated mip-maps from a texture file.

23. Write a program to load and render an animated skinned 3D model.
24. Write a program to capturing and saving an image after a scene has been rendered.
25. Write a program to implement multi-threaded command buffer generation.
26. Write a program to implement instanced mesh rendering.
27. Write a program to demonstrate the use of indirect draw commands.
28. Write a program to demonstrate the use of occlusion query for visibility testing.
29. Write a program to demonstrate the use of query pool objects to gather statistics from different stages of the pipeline
30. Write a program to demonstrate a basic specular BRDF implementation with solid materials and fixed light sources on a grid of objects with varying material parameters.
31. Write a program to demonstrate physical based rendering with image based lighting
32. Write a program to demonstrate physical based rendering with a textured object (metal/roughness workflow) with image based lighting
33. Write a program to demonstrate deferred shading with multiple render targets
34. Write a program to demonstrate multi sampling with explicit resolve for deferred shading
35. Write a program to demonstrate deferred shading with shadows from multiple light sources using geometry shader instancing
36. Write a program to add ambient occlusion in screen space to a 3D scene.
37. Write a program to demonstrate the use of a compute shader with different convolution kernels in realtime.
38. Write a program to demonstrate attraction based compute shader particle system.
39. Write a program to demonstrate compute shader N-body simulation using two passes and shared compute shader memory.
40. Write a program to demonstrate simple GPU ray tracer with shadows and reflections using a compute shader.
41. Write a program to demonstrate compute shader cloth simulation.
42. Write a program to demonstrate compute shader culling and LOD using indirect rendering.
43. Write a program to demonstrate geometry shader (vertex normal debugging).
44. Write a program to demonstrate viewport array with single pass rendering using geometry shaders.
45. Write a program to demonstrate tessellation shader PN triangles.
46. Write a program to demonstrate the use of tessellation to renders a terrain.
47. Write a program to demonstrate minimal headless rendering
48. Write a program to demonstrate minimal headless compute shader
49. Write a program to demonstrate text overlay rendering on-top of an existing scene using a

separate render pass.

50. Write a program to demonstrate font rendering using signed distance fields.

51. Write a program to generate and renders a complex user interface with multiple windows, controls and user interaction on top of a 3D scene.

52. Write a program to demonstrate the basics of full screen shader effects.

53. Write a program to demonstrate bloom effect with full screen shader effects.

54. Write a program to implement multiple texture mapping methods to simulate depth based on texture information (Normal mapping, parallax mapping, steep parallax mapping and parallax occlusion mapping).

55. Write a program to demonstrate the use of a spherical material capture texture array defining environment lighting and reflection information to fake complex lighting.

56. Write a program to demonstrate the use of push descriptors apply the push constants concept to descriptor sets.

57. Write a program to demonstrate the use of the VK_EXT_debug_marker extension to set debug markers, regions and to name Vulkan objects for advanced debugging in graphics debuggers like RenderDoc.

58. Write a program to demonstrate the use of animated gears using multiple uniform buffers

59. Write a program to render a Vulkan demo scene with logos and mascots.

4. CUDA Experiments

1. Write a program to test Asynchronous data transfer using CPU to overlap execution over GPU.
2. Write a program to measure performance using clock functions.
3. Write a program to execute CUDA function into existing CPP application.
4. Write a Program to use CPP function overloading in CUDA API using attribute check.
5. Write a program to demonstrate use of OpenMP for using Multiple GPU.
6. Write a program to demonstrate use of inline PTX assembly language.
7. Write a program to do Matrix Multiplication using CUDA API and also run performance analysis.
8. Write a program to do Matrix Multiplication using high performance CUDA Library CUBLAS.
9. Write a program to do Matrix Multiplication and demonstrate CUDA programming principles and performance analysis.
10. Write a program to demonstrate use of Assert functions in CUDA programming.
11. Write a program to demonstrate use of global memory Atomic function for arithmetic operations.
12. Write a program to create heterogeneous CPU Callbacks for GPU CUDA streams and events.
13. Write a program to take 3D input array and fetch 2D cubemap texture data for each layer and write 3D output array.
14. Write a program to fetch texture from layed 2D texture input.
15. Write a program to demonstrate MPI programming using some calculation done on multiple nodes GPU
16. Write a program to demonstrate multiple memory copy overlap between host and device.
17. Write a program to do some arithmetic operations on multiple GPU and profile it against CPU.
18. Write a program to demonstrate occupancy calculated kernel launch again manual configured.
19. Write a program to demonstrate texture bound to pitch linear memory.
20. Write a program to show printf implementation on CUDA device.
21. Write a program to demonstrate how to create static library and use for compiling CUDA application.
22. Write a program to demonstrate use of CUDA streams to overlap memcopy to host (CPU) memory to improve performance.
23. Write a program to demonstrate write to texture using a simple program of rotating a provided image.
24. Write a program to demonstrate correct use of template using dynamically allocated shared memory arrays.
25. Write a program to demonstrate how to use texture fetches in CUDA.
26. Write a program to demonstrate how to fetch texture in CUDA using kernel launch driver API.

27. Write a program to use vote intrinsic instructions in CUDA kernels.
28. Write a program to use zero memcopy using pinned system memory access.
29. Write a program to provide template for CUDA project.
30. Write a program to use a CUDA runtime template.
31. Write a program to demonstrate unified memory streams access on GPU using OpenMP.
32. Write a program to do simple vector addition.
33. Write a program to do vector addition using driver API for kernel launch.
34. Write a program to measure the memcopy bandwidth of the GPU.
35. Write a program to query properties of CUDA devices.
36. Write a program to query CUDA device properties using kernel launch driver API.
37. Write a program to get peer to peer bandwidth latency.
38. Write a program to demonstrate Bindless Surface/Texture.
39. Write a program to demonstrate Mandelbrot or Julia Fractals set interactively
40. Write a program to demonstrate Marching Cubes Algorithm to extract isosurfaces.
41. Write a program to generate Sine wave using CUDA and generate geometry using OpenGL.
42. Write a program to demonstrate use of 3D textures in CUDA using OpenGL.
43. Write a program to demonstrate volumetric filtering using 3D Texture and Surface writes.
44. Write a program to demonstrate 3D Volumetric rendering with 3D textures.
45. Write a program to demonstrate how to efficiently implement bicubic texture filtering in CUDA.
46. Write a program to uses CUDA to perform a simple bilateral filter on an image and uses OpenGL to display the results.
47. Write a program to use CUDA to perform a simple box filter on an image and uses OpenGL to display the results.
48. Write a program to do 2D convolution using FFT Transformation.
49. Write a program to apply separable convolution filter to 2D signal using Gaussian kernel.
50. Write a program to do texture based 2D convolution using Gaussian kernel.
51. Write a program to demonstrate Discrete Cosine Transform (DCT) for blocks of 8 by 8 pixels
52. Write a program to demonstrate 1D discrete Haar wavelet decomposition.
53. Write a program to demonstrate DirectX Texture Compressor (DXTC) using CUDA.
54. Write a program to implement 64-bin and 256-bin histogram.
55. Write a program to demonstrate Variational optical flow estimation.
56. Write a program to demonstrate two image denoising techniques KNN and NLM.
57. Write a program to post process an image in OpenGL using CUDA
58. Write a program to implement Gaussian blur using recursive method.
59. Write a program to demonstrate CUDA and OpenGL interop for image.

60. Write a program to demonstrate Sobel Edge detection filter.
61. Write a program to implement Stereo Disparity Computation (SAD SIMD Intrinsics).
62. Write a program to evaluate fair call price for a given set of European options under binomial model.
63. Write a program to evaluate fair call and put prices for a given set of European options by Black-Scholes formula.
64. Write a program to evaluate fair call price for a given set of European options using Monte Carlo approach.
65. Write a program to demonstrate a NiederreiterQuasirandom Sequence Generator.
66. Write a program to demonstrate Sobol Quasi-random Number Generator.
67. Write a program to simulate fluid using openGL and CUFFT library.
68. Write a program to simulate N-Body using CUDA.
69. Write a program to Simulate Ocean using CUFFT and OpenGL library.
70. Write a program to simulate large set of particles and their physical interaction over a fixed grid.
71. Write a program to implement high performance method for adding volumetric shadowing to particle systems.
72. Write a program to demonstrate the access speed difference when using aligned and misaligned data structure.
73. Write a program to implement concurrent execution of multiple kernels.
74. Write a program to compute all Eigen values using bisectional algorithm.
75. Write a program to implement Fast Walsh Transform.
76. Write a program to apply time domain progression stencil on a 3D surface.
77. Write a program to demonstrate use of function pointers using Sobel Edge Detection application.
78. Write a program to demonstrate use of recursive computation over interval arithmetic operation.
79. Write a program to demonstrate line of sight algorithm.
80. Write a program to do matrix multiplication using Just in Time (JIT) compilation using PTX code.
81. Write a program to demonstrate merge sort algorithm.
82. Write a program to demonstrate global dynamic allocation using new and delete operator.
83. Write a program to simply demonstrate JIT (Just in Time) compilation using PTX code kernel.
84. Write a program to demonstrate radix sort algorithm using CUDA API with Thrust library.
85. Write a program to implement CUDA Parallel Reduction over large arrays.
86. Write a program to demonstrate scalar product of vector pair.
87. Write a program to demonstrate CUDA Parallel Prefix Sum (Scan).
88. Write a program to demonstrate CUDA segmentation tree thrust library.
89. Write a program to demonstrate CUDA Parallel Prefix Sum with Shuffle Intrinsics (SHFL_Scan).

90. Write a program to implement concurrent CUDA streams using HyperQ.

91. Write a program to demonstrate sorting networks algorithm.

92. Write a program to demonstrate reduction using thread fence intrinsic operation.

93. Write a program to demonstrate CUDA Context Management use for Multi-threading.

94. Write a program to implement Matrix Transpose using different performance algorithm.

95. Write a program to get performance by using batch of CUBLAS API calls.

96. Write a program to perform BoxFilter using NPP library functions.

97. Write a program for conjugate gradient solver using CUBLAS and CUSPARSE Library.

98. Write a program for preconditioned Conjugate Gradient using CUDA libraries.

99. Write a program to implement interoperability between FreeImage and NPP library.

100. Write a program to implement grabcut algorithm using NPP library.

101. Write a program for image histogram equilization using NPP.

102. Write a program to show image segmentation using NPP.

103. Write programs to JPEG encode/decode and resize using NPP.

104. Write a program to perform Monte Carlo estimation of PI (inline PRNG) using CURAND library.

105. Write a program to perform Monte Carlo estimation of PI (inline QRNG) using CURAND library.

106. Write a program to perform Monte Carlo estimation of PI (batch PRNG) using CURAND library.

107. Write a program to perform Monte Carlo estimation of PI (batch QRNG) using CURAND library.

108. Write a program to perform Monte Carlo simulation for single Asian option using CURAND library.

109. Write a program to perform Mersenne Twister GP11213 random number generator using CURAND.

110. Write a program to demonstrate use of random number by generating Random Fog.

111. Write a program to show use of CUBLAS library using CPU GPU test.

112. Write a program to demonstrate 1D-Convolution using CUFFT library.

113. Write a program to solve 2D-Poisson equation using CUFFT library.

Contact us

Registered Office

SIGMA TRAINERS AND KITS
E-113, Jai Ambe Nagar,
Near Udgam School,
Drive-in Road,
Thaltej,
AHMEDABAD-380054. INDIA.

Contact Person

Prof. D R Luhar – Director

Mobile	: 9824001168
Whatsapp	: 9824001168

Phones:

Office	: +91-79-26852427
Factory	: +91-79-26767512
	+91-79-26767648
	+91-79-26767649

Factory

SIGMA TRAINERS AND KITS
B-6, Hindola Complex,
Below Nishan Medical Store,
Lad Society Road,
Near Vastrapur Lake,
AHMEDABAD-380015. INDIA.

E-Mails :

sales@sigmatrainers.com
drluhar@gmail.com