Orientation Plan notes and documents:  
  
1. Introduction:

## AI:

Artificial Intelligence (AI) is the branch of computer science that focuses on building machines and systems capable of performing tasks that normally require human intelligence.

These tasks include:

* Learning from experience (like humans learn from past mistakes)
* Understanding language
* Recognizing patterns (faces, voices, handwriting)
* Solving problems & making decisions
* Adapting to new situations

### Various branches of AI:

#### 1. Machine Learning (ML)

* Definition: AI systems learn patterns from data and improve over time without explicit programming.
* Key Methods: Supervised learning, unsupervised learning, reinforcement learning.
* Examples:
  + Spam filters (email)
  + Fraud detection in banking
  + Product recommendations (Amazon, Netflix)

#### 2. Deep Learning (DL)

* Definition: Subfield of ML using artificial neural networks with multiple layers (mimicking the human brain).
* Strengths: Handles complex data like images, speech, video.
* Examples:
  + Facial recognition (Facebook, iPhones)
  + Self-driving car perception systems
  + Chatbots (like ChatGPT)

#### 3. Natural Language Processing (NLP)

* Definition: Enables machines to understand, interpret, and generate human language.
* Applications:
  + Sentiment analysis (Twitter monitoring)
  + Voice assistants (Siri, Alexa)
  + Language translation (Google Translate)

#### 4. Computer Vision (CV)

* Definition: Allows machines to interpret images and videos.
* Applications:
  + Medical imaging (tumor detection)
  + Traffic surveillance & facial recognition
  + Retail (Amazon Go stores with no checkout counters)

#### 5. Robotics

* Definition: AI combined with physical machines for intelligent action.
* Applications:
  + Manufacturing robots (assembly lines)
  + Delivery robots (Amazon drones)
  + Humanoid robots (Sophia)

#### 6. Expert Systems

* Definition: Rule-based AI systems that simulate human decision-making.
* Applications:
  + Healthcare diagnosis systems
  + IT troubleshooting assistants
  + Legal advisory tools

#### 7. Reinforcement Learning (RL)

* Definition: Systems learn through trial-and-error using rewards and penalties.
* Applications:
  + AlphaGo (defeated human Go champion)
  + Autonomous driving
  + Robotics movement learning

#### 8. Evolutionary Algorithms & Swarm Intelligence

* Definition: Algorithms inspired by nature (evolution, ants, bees, flocks of birds).
* Applications:
  + Logistics optimization (delivery routes)
  + Network traffic optimization
  + Engineering design solutions

### Impact of AI in modern world and different businesses:

#### General Impact

* Increases efficiency & automation.
* Extracts insights from big data.
* Enables personalized services.
* Supports smarter decision-making.

#### Healthcare

* AI-powered diagnostic tools (X-rays, MRI analysis).
* Predicts disease risks & outbreaks.
* Personalized treatment recommendations.
* AI chatbots for patient queries.

#### Finance

* Fraud detection using anomaly detection.
* Algorithmic stock & crypto trading.
* Chatbots for customer support.
* Automated credit scoring & loan approval.

#### Retail & E-commerce

* Product recommendation engines (Netflix, Amazon).
* Smart inventory & demand forecasting.
* AI-powered chatbots for sales support.
* Virtual try-on in fashion & cosmetics.

#### Transportation

* Autonomous cars (Tesla, Waymo).
* AI route optimization (Uber, logistics).
* Smart traffic management in cities.
* Predictive maintenance for airlines.

#### Manufacturing

* Robots on assembly lines.
* Computer vision for quality inspection.
* Predictive maintenance of machinery.
* Supply chain optimization.

#### Marketing & Business

* AI-driven targeted ads.
* Customer segmentation & insights.
* Social media sentiment analysis.
* Automated content generation (AI copywriting tools).

#### Education

* Personalized learning platforms (Duolingo, Khan Academy AI tutors).
* Automated grading.
* Smart assistants for students.
* AI-based plagiarism detection.

#### Entertainment

* Video game AI for NPCs and environments.
* AI-generated music, art, and films.

Personalized movie/song recommendations.

* Virtual influencers & digital actors.

# Machine learning:

AI systems learn patterns from data and improve over time without explicit programming.

## Key Methods:

### Supervised learning:

Definition: Learn from labelled data (input → output known).

Goal: Predict new outcomes.

* Linear Regression → Predicts continuous values (e.g., house prices).
* Logistic Regression → Binary classification (e.g., spam vs. not spam).
* Decision Trees → Rule-based classification.
* Random Forest → Multiple decision trees combined (ensemble).
* Support Vector Machines (SVM) → Classifies data by finding best separation line.
* k-Nearest Neighbors (k-NN) → Classifies based on closest neighbours.
* Neural Networks (ANNs) → Used for complex classification (images, text).

### unsupervised learning:

Definition: Learn from unlabelled data (no outputs given).

Goal: Find hidden patterns, clusters, or reduce complexity.

* K-Means Clustering → Groups data into *k* clusters.
* Hierarchical Clustering → Builds clusters in a tree structure.
* DBSCAN (Density-Based Spatial Clustering) → Detects clusters of varying shapes/densities.
* Principal Component Analysis (PCA) → Reduces dimensionality (feature extraction).
* Autoencoders (Neural Nets) → Learn hidden representations of data.

### reinforcement learning:

Definition: Learning by trial and error using rewards & penalties.

Goal: Optimize actions over time.

* Q-Learning → Learns the value of actions in states.
* SARSA (State-Action-Reward-State-Action) → Similar to Q-learning but updates differently.
* Deep Q-Networks (DQN) → Combines deep learning with Q-learning.
* Policy Gradient Methods (e.g., REINFORCE) → Directly optimize policies.
* Actor-Critic Methods (A3C, DDPG, PPO) → Combine value-based & policy-based methods.

## What kind of problems can we solve using ML?

### Classification Problems

* Assign input into categories.
* Examples:
  + Email spam vs. not spam
  + Disease detection (positive/negative)
  + Image recognition (dog, cat, human)

### Regression Problems

* Predict continuous numeric values.
* Examples:
  + Predicting stock prices
  + Estimating house prices
  + Predicting rainfall

### Clustering Problems

* Group data into meaningful clusters.
* Examples:
  + Customer segmentation in marketing
  + Social media community detection
  + Grouping products based on features

### Anomaly Detection

* Detect unusual patterns in data.
* Examples:
  + Credit card fraud detection
  + Intrusion detection in cybersecurity
  + Fault detection in manufacturing

### Recommendation Systems

* Suggest items based on past behaviour.
* Examples:
  + Netflix suggesting movies
  + Amazon product recommendations
  + Spotify song playlists

### Reinforcement Learning Applications

* Robotics → Teaching robots to walk or grasp objects.
* Gaming → AlphaGo, chess-playing AI.
* Self-driving cars → Learn traffic rules and driving safety.

## What is deep learning :

Definition:  
 Deep Learning is a subfield of Machine Learning that uses Artificial Neural Networks (ANNs) with many layers (deep networks) to automatically learn complex patterns from large amounts of data.

Key Idea:  
 Instead of manually designing features, DL automatically extracts features from raw data (images, text, audio).

Why “Deep”?  
 Because neural networks have multiple layers (hidden layers) that process data step by step.

### Daily Life Examples:

* + Face unlock on smartphones
  + Google Translate language translation
  + Self-driving car object detection
  + Chatbots like ChatGPT

### Various DL algorithms:

1. Artificial Neural Networks (ANNs)

* Basic structure inspired by human brain neurons.
* Used for general prediction tasks (classification, regression).

#### 2. Convolutional Neural Networks (CNNs)

* Specialized for image and video data.
* Extract features like edges, shapes, textures.
* Applications: Image classification, face recognition, medical imaging.

#### 3. Recurrent Neural Networks (RNNs)

* Designed for sequential data (time series, text, speech).
* Keeps track of previous inputs with memory.
* Applications: Speech recognition, text generation, stock prediction.

#### 4. Long Short-Term Memory (LSTM) & GRU

* Advanced forms of RNNs that solve the “vanishing gradient” problem.
* Applications: Language translation, chatbots, handwriting recognition.

#### 5. Generative Adversarial Networks (GANs)

* Two networks (Generator & Discriminator) compete with each other.
* Applications:
  + Generating realistic images
  + Deepfake videos
  + Data augmentation for training ML models.

#### 6. Transformers

* State-of-the-art architecture for NLP & beyond.
* Uses self-attention mechanism to process sequences efficiently.
* Applications:
  + Large Language Models (ChatGPT, Google Bard).
  + Machine translation.
  + Summarization & Q&A systems.

#### 7. Deep Reinforcement Learning

* Combines neural networks + reinforcement learning.
* Applications:
  + AlphaGo (beating world champion in Go).
  + Robotics control.
  + Autonomous vehicles.

## Problems that can be solved by Deep learning:

### Computer Vision Problems

* Object detection, facial recognition, medical image diagnosis.
* Self-driving cars (detect pedestrians, traffic lights).

### Natural Language Processing Problems

* Machine translation (Google Translate).
* Chatbots & virtual assistants (ChatGPT, Siri, Alexa).
* Sentiment analysis on social media.

### Speech & Audio Processing

* Speech-to-text (Google Voice Typing).
* Voice assistants.
* Music generation & sound classification.

### Generative Tasks

* Creating new images, videos, or text.
* Deepfakes, AI art (DALL·E, MidJourney).
* Synthetic data generation for training.

### Healthcare

* Detecting diseases from X-rays/MRIs.
* Predicting patient outcomes.
* Drug discovery.

### Gaming & Robotics

* AI agents mastering complex games (Dota, StarCraft).
* Robots learning to walk, grasp, and perform tasks.

# Computer Vision:

Definition:  
 Computer Vision (CV) is a field of AI that enables machines to interpret, understand, and analyse visual information (images & videos) from the real world.

Goal:  
 Replicate human vision—detect, recognize, and understand objects/scenes in digital media.

Key Idea:  
 Convert images/videos → into numerical representations → apply algorithms → extract meaning.

### Daily Life Examples:

* Face unlock on smartphones
* Google Photos automatically tagging people ️
* Self-driving cars detecting pedestrians
* Medical imaging for disease detection

## Various Computer Vision Algorithms:

### Traditional (Before Deep Learning)

* Edge Detection (Sobel, Canny) → Detect edges in images.
* Template Matching → Find a small pattern in a larger image.
* HOG (Histogram of Oriented Gradients) → Used for object detection.
* SIFT (Scale-Invariant Feature Transform) → Detect keypoints/features in images.
* SURF (Speeded-Up Robust Features) → Faster alternative to SIFT.
* Viola-Jones Algorithm → Early face detection method.

### Deep Learning Based

* Convolutional Neural Networks (CNNs) → Core of modern CV (image classification, object detection).
* YOLO (You Only Look Once) → Real-time object detection.
* Faster R-CNN / Mask R-CNN → Object detection + segmentation.
* ResNet, VGG, Inception → Image classification architectures.
* U-Net → Image segmentation (used in healthcare).
* Vision Transformers (ViT) → Transformer models adapted for image tasks.
* GANs for Vision → Generate realistic images, style transfer, data augmentation.

## What Problems Can We Solve with Computer Vision?

### Image Classification

* Identifying objects in images.
* Examples: Cat vs Dog classifier, diagnosing diseases from X-rays.

### Object Detection

* Locating & labeling objects in images/videos.
* Examples: Self-driving cars (pedestrians, traffic lights).

### Image Segmentation

* Dividing image into regions/objects.
* Examples: Tumor segmentation in medical scans.

### Face Recognition & Verification

* Unlocking smartphones, security systems.
* Social media auto-tagging.

### Optical Character Recognition (OCR)

* Extracting text from images/scans.
* Examples: Google Lens, scanning documents.

### Action & Gesture Recognition

* Video surveillance (detecting suspicious activity).
* Human-computer interaction via gestures.

### Healthcare Applications

* Detecting pneumonia in chest X-rays.
* Identifying cancer in biopsy images.

### Retail & Industry

* Smart checkout systems (Amazon Go).
* Quality inspection in manufacturing.

### Agriculture

* Detecting crop diseases from leaf images.
* Estimating crop yield.

# LLMS (Large Language Models):

## 1. What is an LLM?

* A Large Language Model (LLM) is a type of artificial intelligence model designed to understand, generate, and interact with human language.
* It is a special application of deep learning that uses transformer architectures and is trained on massive amounts of text data.

### LLMs can:

* + Understand natural language (NLU).
  + Generate human-like responses (NLG).
  + Perform reasoning, summarization, translation, coding, and more.

## Core Architectures of LLMs

LLMs are primarily built on the Transformer architecture (2017, "Attention is All You Need").

### Main Components:

1. Tokenization – Text is broken down into tokens (words, subwords, or characters).
2. Embeddings – Each token is represented as a high-dimensional vector.
3. Attention Mechanism – The model decides which words in a sentence should focus on each other.
   1. Example: In "The cat sat on the mat", the word "cat" relates strongly to "sat".
4. Feed-forward Neural Networks – Transform input embeddings to capture deep contextual relationships.
5. Stacking Layers – Transformers have multiple encoder/decoder layers (sometimes hundreds in LLMs).
6. Output Layer – Generates the probability distribution of the next token.

#### Popular Architectures:

* GPT (Generative Pretrained Transformer) – Decoder-only model, good at text generation.
* BERT (Bidirectional Encoder Representations from Transformers) – Encoder-only model, great at understanding tasks like classification & QA.
* T5 (Text-to-Text Transfer Transformer) – Encoder-decoder model, designed to convert any NLP task into text-to-text format.
* LLaMA, Falcon, Mistral, Claude, Gemini – Other advanced modern LLM families.

### Problems that can be sorted using LLMS:

#### **Natural Language Processing (NLP) Tasks**

* **Text classification** (spam detection, sentiment analysis).
* **Named Entity Recognition (NER)** (extract names, dates, places).
* **Question answering** (like ChatGPT answering queries).
* **Machine translation** (English → French, Urdu → English).
* **Summarization** (condense long text into short summaries).

#### **Content Generation**

* Essay and article writing.
* Code generation (e.g., GitHub Copilot).
* Creative writing (poems, stories, marketing copy).

#### **Conversational AI**

* Virtual assistants (ChatGPT, Siri, Alexa).
* Customer service chatbots.

#### **Reasoning & Knowledge Applications**

* Chain-of-thought reasoning for solving math/logic.
* Legal/medical assistants.
* Scientific research support.

#### **Multimodal Applications (new trend)**

* Text + Image (e.g., GPT-4V, Gemini, Claude 3).
* Text + Speech.
* Text + Video understanding.

### Why was there a need for the Transformer model?

Before **Transformers** (2017, *Vaswani et al., Attention Is All You Need*), most NLP systems relied on:

* **RNNs (Recurrent Neural Networks)**
* **LSTMs (Long Short-Term Memory networks)**
* **GRUs (Gated Recurrent Units)**

These worked for sequential text but had **major limitations**:

#### 1. Limitations of RNNs/LSTMs

* **Sequential bottleneck**:
  + RNNs process text word by word. This makes training and inference **slow**, especially for long sentences/documents.
* **Difficulty capturing long-range dependencies**:
  + Example: In the sentence *“The book that I read yesterday was amazing”*, RNNs often struggle to link “book” with “amazing” because the words are far apart.
* **Vanishing/exploding gradients**:
  + During training, gradient updates either shrink (vanish) or grow uncontrollably (explode), making it hard to learn from long sequences.
* **Limited parallelization**:
  + RNNs must process input step by step (can’t train in parallel easily). This made large-scale training slow.

#### 2. Why Transformers Were Introduced

Transformers solved these issues by introducing the **Self-Attention Mechanism**.

* **Parallelization**:
  + Unlike RNNs, Transformers process **all words at once** instead of sequentially. This speeds up training drastically.
* **Long-range dependency handling**:
  + Self-attention allows the model to directly connect **any word with any other word**, no matter how far apart they are in the sentence.
* **Scalability**:
  + Works well with massive datasets and huge model sizes (billions of parameters).
* **Better performance**:
  + Achieved **state-of-the-art** results in translation, summarization, and text generation almost immediately after introduction.

#### 3. Real-World Example

* Old RNN/LSTM translation:
  + Might fail in long sentences → “The cat that chased the mouse… fell asleep.” (model might lose the connection between *cat* and *fell asleep*).
* Transformer translation:
  + Uses attention to link *cat* with *fell asleep* directly, regardless of distance in the sentence.

# Explore Agentic AI:

## **What is an Agent in AI?**

An **AI Agent** is a system that:

* **Perceives** its environment (via input data, sensors, APIs, etc.),
* **Thinks/Plans** using reasoning or models,
* **Acts** to achieve goals (by executing actions, making decisions, or interacting with software/world).

**Agentic AI** = AI systems designed to **act autonomously**, not just give outputs like ChatGPT, but to:

* Make decisions
* Plan multi-step tasks
* Call tools / APIs
* Adapt based on feedback

## **Core Architectures / Components of Agentic AI**

An Agentic AI usually has the following parts:

### Environment

* 1. The world the agent interacts with (real world, web, or a software system).

### Perception (Input / Sensors)

* 1. How the agent understands the world (data, user queries, APIs, databases).

### Reasoning & Planning

* 1. Brain of the agent.
  2. Uses ML models, LLMs, or symbolic reasoning to plan next steps.
  3. Example: Chain-of-thought reasoning, planning algorithms.

### Memory

* 1. Stores past interactions and knowledge (short-term & long-term memory).
  2. Helps agent learn from experience.

### Decision-Making (Policy/Controller)

* 1. Chooses what action to take next based on goals and context.

### Action / Actuators

* 1. Executes the chosen action:
     + Sending an email
     + Booking a ticket
     + Calling an API
     + Moving a robot arm

### Feedback Loop (Learning)

* 1. Evaluates outcome of its actions and adjusts behavior.
  2. Often uses **reinforcement learning** or reward signals.

## **Core Architectures in Practice**

* **Reactive Agents** → Respond immediately (e.g., spam filter, thermostat).
* **Deliberative Agents** → Plan before acting (e.g., chess-playing AI).
* **Hybrid Agents** → Combine both.
* **LLM-powered Agents** → Use LLMs (GPT, Claude, etc.) for reasoning + tool use

## **What Sort of Problems Can Agentic AI Solve?**

### Digital/Business Automation

* Customer support chatbots that resolve issues end-to-end
* Virtual assistants (e.g., scheduling meetings, booking tickets)

### Decision-Making

* Stock trading bots
* Supply chain optimization

### Robotics

* Self-driving cars
* Factory robots
* Drone delivery

#### Healthcare

* AI agents for medical diagnosis assistance
* Personalized treatment planning

#### Research & Knowledge Work

* AI agents that read papers, summarize, and propose new experiments
* Coding agents that debug and write software