**Objective**

Develop a full-stack solution for a Connect 4 game bot that includes training machine learning models, deploying them in a cloud environment, and creating an interactive web application for users to play against the bot. The project will integrate machine learning, cloud computing, and front-end development to deliver a complete user experience.

**Tasks**

1. **Dataset Creation Using Monte Carlo Tree Search (MCTS)**
   * Generate a diverse dataset of Connect 4 board positions and corresponding best moves using MCTS.
   * Simulate games with MCTS playing against itself, ensuring data quality by excluding redundant or inconsistent moves.
   * Use the dataset to replicate human-like expertise for the bot’s decision-making process.
2. **Neural Network Training**
   * Train two machine learning models:
     1. **Convolutional Neural Network (CNN):** Treat the Connect 4 board as a 2D grid and optimize network architecture for classification.
     2. **Transformer Network:** Apply attention-based mechanisms to the board’s state for predicting optimal moves.
   * Evaluate model performance by:
     1. Testing against a less sophisticated MCTS bot.
     2. Measuring accuracy on validation datasets.
   * Compare architectures to determine the better-performing model.
3. **Web Application Development**
   * Build an interactive web page using **Anvil** that:
     1. Describes the training process, challenges, and performance of the neural networks.
     2. Allows users to select a bot (CNN or Transformer) and play against it.
     3. Provides a restart game feature and user authentication.
   * Ensure the application is user-friendly and visually appealing.
4. **Backend Deployment**
   * Host the trained models on **AWS Lightsail** for backend integration with the web application.
   * Dockerize the backend code to ensure compatibility across environments.
   * Connect the Anvil front-end with AWS-hosted services to access Python packages like TensorFlow.
5. **Submission Requirements**
   * Submit a fully functional application with:
     1. AWS-hosted backend code and Anvil front-end link.
     2. A detailed analysis of the model-building process, including success cases and limitations.
     3. User authentication (e.g., pre-created credentials for evaluation).