

You have to implement one DL project during the semester which include following criteria.

Objective :

Create Complete DL project using Flask/ Fast API of given problem statement with neat and clean UI. The project includes a clean UI for data input and prediction visualization.

Project should Include following:

Objective	Title	Description	Execution Period	Evaluation Period
1	Problem Definition & Dataset Acquisition	<ul style="list-style-type: none"> Define the Deep Learning task (e.g., Classification, Object Detection, NLP). Collect unstructured data (Images, Audio, or Text). Analyze class balance and data quality. 	19/01 to 24/01	26/01 to 31/01
2	Data Pre-processing & Augmentation	<ul style="list-style-type: none"> Images: Resize, Rescale, Normalization. Text/Audio: Tokenization, Padding, Spectrogram conversion. Augmentation: Apply rotation, flipping, zooming to increase dataset size. Split into Train/Validation/Test sets. 	26/01 to 31/01	02/02 to 7/02
3	Model Architecture Design & Training	<ul style="list-style-type: none"> Design a custom Neural Network architecture (ANN, CNN, or RNN) from scratch. Define layers (Conv2D, MaxPool, Dense, Dropout). Compile model with appropriate Optimizer (Adam/SGD) and Loss Function. Train the model for standard epochs. 	02/02 to 7/02	09/02 to 14/02
4	Evaluation & Hyperparameter Tuning	<ul style="list-style-type: none"> Plot Loss vs. Epoch and Accuracy vs. Epoch graphs. Detect Overfitting/Underfitting and apply Regularization (Dropout, L1/L2). Tune Hyperparameters (Learning Rate, Batch Size, Number of Layers). 	09/02 to 14/02	16/02 to 21/02

		<ul style="list-style-type: none"> • Generate Confusion Matrix and Classification Report. 		
5	Application Interface (Frontend)	<ul style="list-style-type: none"> • Learn to set up a Deep Learning inference pipeline in Flask/Streamlit. • Create a web form to accept user input (Image Upload or Text Input). • Pre-process user input to match model requirements before prediction. 	16/02 to 21/02	23/02 to 28/02
6	Advanced Optimization	<ul style="list-style-type: none"> • Transfer Learning: Implement VGG16, ResNet, or MobileNet for better accuracy. • Compare Custom Model vs. Pre-trained Model results. • Save the best model (.h5 or .pt file) for deployment. 	23/02 to 28/02	02/03 to 07/03
7	Backend Integration & Deployment	<ul style="list-style-type: none"> • Connect the Flask/Streamlit frontend with the saved model. • Handle real-time prediction requests. • Deploy the final application to a free hosting platform (e.g., Hugging Face Spaces, Render, or Railway). 	02/03 to 07/03	09/03 to 14/03
8	End	Final Evaluation of Project	09/03 to 14/03	-

Evaluation of Objective:

Evaluation of project will be done **in regular lab**. For every one objective have **one marks** for end semester evaluation.

Grade	Performance Level	Description
A	Very Good	Everything is perfect. The student clearly understands the project, and their work is neat, organized, and complete.
B	Good	Great effort. They understand the main parts of the project, but there are still a few small things that need to be fixed or improved.
C	Average	Just doing the work to get it done. They are finishing tasks but don't really understand how the code works or why they are doing it.

Z	Poor	Not doing the work. They are not following the project rules and haven't shown any interest in starting.
---	------	--

Use Following dataset for ML task:

Sr.	Project Title	Category	Library
1	Face Mask Detector https://www.kaggle.com/datasets/andrewmvd/face-mask-detection/data	Vision	PyTorch
2	Traffic Sign Recognition https://idd.insaan.iiit.ac.in/dataset/details/	Vision	PyTorch, Ultralytics
3	Face Moment Monitoring for Exam(Proctoring) https://www.kaggle.com/datasets/dhruv413/mpiigaze https://ibug.doc.ic.ac.uk/resources/300-W/	Vision	OpenCV, TensorFlow, MediaPipe
4	Vehicle Log System (License Plate Recognition) https://www.kaggle.com/datasets/saisirishan/indian-vehicle-dataset	Vision	EasyOCR, Tesseract, YOLOv8
5	Music Genre Classification https://www.kaggle.com/datasets/carlthome/gtzan-genre-collection	Audio	Librosa, PyTorch
6	Chat with your PDF	Text	LangChain, ChromaDB, HuggingFace, Ollama
7	Generate MCQ from PDF	Text	LangChain, ChromaDB, HuggingFace, Ollama, PyPDF2
8	Stock Price Forecaster Yahoo Finance API	Stock	PyTorch,

9	Synthetic Data Generator for Traffic Scenarios	Image to image	PyTorch, GANs.
10	Website Creator using Ollama	Text	Ollama, Streamlit, LangChain
11	Crowd Anomaly Detection Detect violence, stampedes, or weapon detection in real-time video feeds from public places.	Vision	YOLOv8, OpenCV, PyTorch.
12	Pothole Detection System Build a system that detects potholes from dashcam footage or user-uploaded images and geo-tags them on a map for authorities.	Vision	Object Detection (YOLO), Google Maps API, Flask Backend.