INTERNSHIP PROJECT PHASE (2 WEEKS)

PROJECT GUIDELINES

- **Choose any 1 or 2 projects from the given list.**
- **You are free to improvise — take the given project as a base and modify it as you like.**
- **You can use any tools, technologies, or steps you're comfortable with — there are no restrictions.**
- **Focus and work sincerely so that you have complete clarity and can explain the project confidently in interviews.**
- **Go through the Top 50 Interview Questions for your domain (attached at the end).**
- **Update your project status regularly when the Google Form is shared in group.**
- **while working on the project YOU CAN CHOOSE ANY DATASET RELAVENT TO THE PROJECT.**

.

After project completion, prepare a 1–2 page report in PDF format, containing:

- Introduction
- Abstract
- Tools Used
- Steps Involved in Building the Project
- Conclusion

◆ Note: Report must not exceed 2 pages.

DEAR INTERNS,

YOU HAVE TO UPDATE STATUS OF YOUR PROJECT EVERY 3 OR 4 DAYS ONCE WHEN THE UPDATION LINK IS SHARED IN THE GROUP.

Final Project Submission Date and Guidelines :

23 june 2025: Submission of your final project GitHub repository link with all deliverables and the project report.

If you are doing more than one project put all projects in same repository and prepare report for any one project

Final submission links will be shared later.

! READ ALL THE GUIDELINES CAREFULLY !

LIST OF PROJECTS

1. Face Mask Detection with Live Alert System

Objective: Detect if people are wearing face masks in real-time using a webcam.

Tools: Python, OpenCV, TensorFlow/Keras, Flask, Haar Cascades

Mini-Guide:

- Collect or use Kaggle dataset of masked/unmasked faces
- Preprocess images (resize, grayscale)
- Train CNN model using Keras
- Integrate model with OpenCV video stream
- Use Haar Cascades for face detection
- Add logic to alert when no mask is detected
- Deploy with Flask (optional)

Deliverables: Trained model, real-time detection script, short video demo, GitHub repo

2.AI Virtual Career Counsellor

Objective: Build an NLP chatbot that recommends career paths based on interests.

Tools: Python, NLTK, Rasa, Streamlit

Mini-Guide:

- Create intents (e.g., tech, arts, commerce)
- Use NLTK to preprocess text input
- Train Rasa chatbot with sample dialogues
- Add logic for recommending careers based on keywords
- Create frontend using Streamlit
- Test with real user input
- Deploy via Streamlit Cloud

Deliverables: Rasa project folder, frontend UI, sample interaction video

3.Stock Price Trend Prediction with LSTM

Objective: Predict future stock prices using past trends.

Tools: Python, Keras, Pandas, Matplotlib, Yahoo Finance API

Mini-Guide:

- Fetch data using yfinance API
- Normalize and prepare data
- Build LSTM model using Keras
- Train and validate model
- Plot predictions vs actual
- Integrate moving average & RSI indicators
- Optional: Deploy dashboard with Streamlit

Deliverables: Jupyter notebook, model weights, graphs, Streamlit link

4. AI-Powered Resume Ranker

Objective: Rank resumes for a job profile using NLP techniques.

Tools: Python, SpaCy, Sklearn, Flask

Mini-Guide:

- Extract text from PDF resumes
- Preprocess text using SpaCy
- Vectorize using TF-IDF
- Create a scoring algorithm based on job description keywords
- Rank candidates
- Build a web UI to upload resumes and show ranks
- Add download option for HR reports

Deliverables: Flask app, scoring algorithm, UI, sample outputs

5.Plant Disease Detection from Leaf Images

Objective: Classify plant diseases using image recognition.

Tools: Python, CNN (Keras), TensorFlow, OpenCV

Mini-Guide:

- Use PlantVillage dataset
- Resize & normalize images
- Train CNN on 4-5 classes of diseases
- Evaluate model accuracy
- Create GUI to upload leaf image
- Predict & show results
- Deploy model using Streamlit

Deliverables: Trained model, GUI app, dataset used, video demo

6.News Article Classification (Fake/Real)

Objective: Classify news articles as fake or real using NLP.

Tools: Python, Sklearn, Pandas, NLTK

Mini-Guide:

- Collect labeled dataset from Kaggle
- Clean text using NLTK
- Vectorize with TF-IDF
- Train Logistic Regression/Naive Bayes model
- Evaluate metrics (F1, accuracy)
- Create Streamlit interface for input
- Display prediction & explanation

Deliverables: Jupyter notebook, trained model, live web demo

7. Real-Time Sign Language Recognition
Objective: Convert hand gestures to text in real-time.
Tools: Python, OpenCV, MediaPipe, TensorFlow
Mini-Guide:

- Use MediaPipe for hand tracking
- Label common signs (A–Z or 10 signs)
- Train CNN on hand gesture images
- Predict gesture from live video
- Show corresponding letter/text
- Combine letters into words
- Optional: Text-to-speech output

Deliverables: Model code, webcam script, demo video

8.Movie Recommendation System
Objective: Suggest movies based on user preferences using ML.
Tools: Python, Pandas, Sklearn, Streamlit
Mini-Guide:

- Use MovieLens dataset
- Preprocess and clean data
- Build collaborative filtering model
- Integrate content-based filtering using genres
- Design UI to take input preferences
- Return top 5 recommendations
- Optional: Add sentiment-based filtering using reviews

Deliverables: Notebook, Streamlit UI, recommendation logic

9. AI Chatbot for Mental Health Support
Objective: Provide emotional support using a simple chatbot.
Tools: Python, Transformers (Hugging Face), Flask
Mini-Guide:

- Fine-tune a small conversational model (DialoGPT or BlenderBot)
- Add basic NLP filters for offensive input
- Script empathetic dialogues
- Build a Flask API
- Create frontend with HTML/CSS or Streamlit
- Deploy on Render or Replit
- Add log feature for user sessions

Deliverables: Trained model, API backend, frontend UI

10. Fraud Detection in Credit Card Transactions
Objective: Identify fraudulent transactions using anomaly detection.
Tools: Python, Scikit-Learn, XGBoost, Pandas
Mini-Guide:

- Use Kaggle credit card dataset
- Preprocess & balance the dataset
- Apply Isolation Forest & Local Outlier Factor
- Train XGBoost classifier
- Plot ROC curve
- Create a dashboard for input/prediction
- Deploy as web app

Deliverables: Jupyter notebook, Streamlit/Flask UI, confusion matrix

11. Music Genre Classification
Objective: Predict music genre from audio file.
Tools: Python, Librosa, Keras, CNN
Mini-Guide:

- Load GTZAN dataset
- Extract MFCC features
- Build CNN using Keras
- Train and validate on 80/20 split
- Create script to upload audio and classify
- Show genre with confidence score
- Optional: Add spectrogram visualization

Deliverables: Notebook, trained model, audio prediction demo

12.AI Virtual Personal Fitness Coach
Objective: Detect workout pose and count reps using computer vision.
Tools: Python, OpenCV, MediaPipe, Streamlit
Mini-Guide:

- Detect body landmarks using MediaPipe Pose
- Track joint angles
- Add logic to count reps
- Visualize movement feedback
- Build a real-time app with Streamlit
- Add rep tracker and timer
- Optional: Save workout logs

Deliverables: Code, app link, video walkthrough

13. Human Emotion Detection from Voice

Objective: **Detect speaker emotion using audio input.**

Tools: **Python, Librosa, Sklearn, Streamlit**

Mini-Guide:

1. **Use RAVDESS dataset**
2. **Extract features (MFCC, Chroma, etc.)**
3. **Train classifier (SVM/Random Forest)**
4. **Build UI to record voice**
5. **Run prediction and display emotion**
6. **Optionally add graph of emotional trend**
7. **Save session data**

Deliverables: **Trained model, emotion prediction demo, UI**

14. AI Writer with Text Summarization

Objective: **Generate summaries from large articles using NLP.**

Tools: **Python, Hugging Face Transformers, Flask**

Mini-Guide:

1. **Use BART or T5 summarization model**
2. **Build text input interface**
3. **Process long articles**
4. **Run model and return summary**
5. **Add word count and readability score**
6. **Optional: Save history of summaries**
7. **Deploy using Flask**

Deliverables: **Web app, model integration, sample outputs**

15.AI Dungeon Story Generator

Objective: **Create interactive fantasy stories using generative AI.**

Tools: **Python, GPT-2/GPT-Neo, Streamlit**

Mini-Guide:

1. **Load pretrained GPT-2 from Hugging Face**
2. **Build prompt-based input** 3. **Generate story continuation** 4. **Show multiple continuations**
5. **Save story as text file** 6. **Add genre selection (fantasy, mystery, etc.)** 7. **Deploy via Streamlit**

Deliverables: **Streamlit app, interactive story samples, codebase**

❗TOP 50 INTERVIEW QUESTIONS❗

I. GENERAL AI & ML CONCEPTS

- What is the difference between AI, Machine Learning, and Deep Learning?
- What are the main types of Machine Learning?
- Explain the difference between supervised and unsupervised learning.
- What is overfitting and underfitting?
- How do you evaluate a machine learning model?
- What is the bias-variance trade-off?
- Explain cross-validation and why it's important.
- What are precision, recall, F1-score, and accuracy?
- What is the difference between classification and regression?
- What are some real-world applications of AI/ML?

II. DATA PREPROCESSING & FEATURE ENGINEERING

How do you handle missing data in a dataset?
What is normalization and standardization?
What is one-hot encoding? When would you use it?
How do you handle categorical variables?
What is feature selection and why is it important?
What is dimensionality reduction? Explain PCA.
What is the curse of dimensionality?
What is feature scaling and when should it be applied?
How do you deal with imbalanced datasets?
Explain the role of EDA (Exploratory Data Analysis) in ML.

III. MACHINE LEARNING ALGORITHMS

- How does the Decision Tree algorithm work?
- What is the difference between bagging and boosting?
- Explain how the K-Nearest Neighbors algorithm works.
- What is the intuition behind Support Vector Machines?
- How does Naive Bayes classifier work?
- What is the difference between Random Forest and XGBoost?
- How do gradient descent and stochastic gradient descent differ?
- Explain logistic regression and where it's used.
- What are ensemble models?
- What is the difference between L1 and L2 regularization?

Elevate Labs

IV. DEEP LEARNING & NEURAL NETWORKS

- What is a perceptron?
- How do activation functions like ReLU, Sigmoid, and Tanh work?
- What are epochs, batch size, and learning rate?
- What is the vanishing gradient problem?
- What is the difference between CNN and RNN?
- How does an LSTM work and where is it used?
- What are convolutional layers in CNN?
- What is transfer learning?
- What is dropout and why is it used?
- How do you prevent overfitting in deep learning models?

V. NATURAL LANGUAGE PROCESSING

- What is tokenization in NLP?
- How do word embeddings like Word2Vec or GloVe work?
- What is the difference between stemming and lemmatization?
- What is TF-IDF and why is it used?
- What are Transformers in NLP?

VI. MODEL DEPLOYMENT & MLOps

- How do you save and load a trained model in Python?
- What is model drift and how do you monitor it?
- How do you deploy a machine learning model as an API?
- What are common tools for deploying ML models (Flask, Docker, Streamlit)?
- Explain the CI/CD pipeline in MLOps.