



Recommended Skeleton for Sign2Sounds

```
SIGN2SOUND_TeamName/
├── README.md
├── requirements.txt
├── LICENSE
└── .gitignore

# ★★ MANDATORY - Main documentation
# ★★ MANDATORY - All dependencies
# Recommended (MIT/Apache 2.0/GPL)
# Recommended

├── data/
│   ├── README.md
│   ├── processed/
│   │   └── statistics.txt
│   #
# Dataset information
# ★★ MANDATORY - Dataset sources & instructions
# Optional - Small preprocessed samples
# Dataset statistics (samples, classes, split)

├── preprocessing/
│   ├── preprocess.py
│   ├── augmentation.py
│   ├── extract_features.py
│   └── README.md
#
# ★★ MANDATORY - Data preprocessing
# Main preprocessing script
# Data augmentation functions
# Feature extraction (frames/landmarks)
# Preprocessing pipeline documentation

├── features/
│   ├── hand_landmarks.py
│   ├── pose_estimation.py
│   ├── facial_features.py
│   ├── feature_utils.py
│   └── README.md
#
# ★★ MANDATORY - Feature extraction
# Hand landmark detection
# Body pose extraction
# Facial expression capture (if used)
# Utility functions
# Feature extraction methods

└── models/
    └── model.py
#
# ★★ MANDATORY - Model architecture
# Main model architecture
```

```
└── custom_layers.py          # Custom layers/attention (if any)
└── loss.py                  # Custom loss functions (if used)
└── README.md                # Model architecture documentation

└── training/                # ★★ MANDATORY - Training pipeline
    ├── train.py              # ★★ Main training script
    ├── config.yaml / hyperparams.yaml # Hyperparameters configuration
    ├── callbacks.py           # Training callbacks (if used)
    ├── evaluate.py            # Model evaluation script
    └── README.md              # Training instructions

└── inference/               # ★★ MANDATORY - Inference & demo
    ├── infer.py              # Single inference script
    ├── realtime_demo.py      # Real-time demonstration
    ├── tts.py                 # Text-to-speech module
    ├── utils.py               # Helper functions
    └── README.md              # Inference usage instructions

└── ui/                      # Optional - Only if UI exists
    ├── app.py / index.html   # Main UI application
    ├── static/               # CSS, JS, images
    ├── templates/             # HTML templates (if web app)
    └── README.md              # UI setup instructions

└── notebooks/                # Recommended - Exploratory work
    ├── 01_data_exploration.ipynb # Dataset analysis
    ├── 02_model_experiments.ipynb # Model experimentation
    ├── 03_results_visualization.ipynb # Results analysis
    └── README.md                # Notebook descriptions

└── results/                  # ★★ MANDATORY - Performance results
    ├── metrics.json           # All performance metrics
    ├── confusion_matrix.png   # ★★ Confusion matrix visualization
    ├── loss_curves.png        # ★★ Training/validation loss
    ├── accuracy_curves.png   # ★★ Training/validation accuracy
    ├── per_class_performance.csv # Per-class metrics
    ├── training_log.txt       # Complete training logs
    └── sample_outputs/         # Sample predictions
        ├── sample_1.png
        ├── sample_2.png
        └── predictions.txt

└── checkpoints/              # Model weights
    ├── best_model.h5 / best_model.pth # Best model checkpoint
```

```
├── final_model.h5 / final_model.pth # Final model
└── README.md                      # ★★ MANDATORY - Download links if >100MB

├── docs/                           # ★★ MANDATORY - Documentation
│   ├── architecture_diagram.png    # ★★ Model architecture visualization
│   ├── system_pipeline.png         # ★★ End-to-end system flow
│   ├── technical_report.pdf       # ★★ Complete technical report
│   ├── dataset_preprocessing.md   # Preprocessing details
│   └── training_details.md        # Training procedure details

├── tests/                          # Recommended - Unit tests
│   ├── test_preprocessing.py
│   ├── test_model.py
│   └── test_inference.py

└── scripts/                        # Optional - Utility scripts
    ├── download_datasets.sh
    ├── setup_environment.sh
    └── run_all.sh
```