SUMMARY OF THE PROJECT

METHODOLOGY AND WORKING

- **Database Schema Design (MySQL)**
- **Tables to Create**:
  - **Models Table**: Stores model-specific details (ID, name, type, framework, etc.).
  - **Training Table**: Stores training dataset, hyperparameters, duration, etc.
  - **Metrics Table**: Stores model performance metrics (accuracy, loss, AUC, etc.).
  - **Versioning Table**: Tracks versions of the model (each time a model is retrained, a new version is logged).
  - **Deployment Table**: Logs deployment details (date, environment, endpoints, etc.).

- **Inserting Metadata**
- **Before Training**: Insert model and hyperparameter configurations into the database.
- **After Training**: Insert performance metrics, training duration, dataset used, and the trained model file path or location.
- **On Model Versioning**: Update version history and track changes in the model and parameters.
- **.Querying Metadata**
- **Accessing Model Details**: Retrieve models based on ID, framework, or type.
- **Performance Monitoring**: Retrieve and compare historical performance metrics.
- **Version History**: Track which versions performed better or are deployed in specific environments.
- **Deployment Tracking**: Monitor the production status and rollback history.

- **Best Practices for Metadata Management**
- **Standardization**: Define a standardized format for logging metadata.
- **Automation**: Automate the logging process during training and deployment.
- **Version Control**: Ensure that each model version has associated metadata for reproducibility.
- **Security**: Secure access to metadata for audit purposes and compliance.
- **Integration with ML Pipelines**
- **Logging**: Integrate metadata management with your ML pipeline (e.g., during data preprocessing, model training, evaluation).
- **Traceability**: Ensure that every experiment is logged with all relevant metadata to support debugging, auditing, and collaboration.