PHASE -2 DOCUMENT

Student Name: **M.Narmatha**

Register No: **623323205020**

Institution: **Vetri Vinayaha College of Engineering and Technology**

Department: **B.Tech Information Technology**

Date of Submission: **07/05/2025**

1. Problem Statement

The goal is to develop an AI system capable of recognizing handwritten digits (0–9) with high accuracy. This enhances applications in areas like postal mail sorting, bank check processing, and digital form reading.

1. Project Objectives

Build a robust digit classification model using deep learning.

• Preprocess and analyze the MNIST dataset.

• Evaluate model performance and optimize accuracy.

• Visualize model predictions and learning behavior.

2. Flowchart of the Project Workflow

Start

↓

Data Collection (MNIST)

↓

Data Preprocessing

↓

Exploratory Data Analysis

↓

Feature Engineering (if applicable)

↓

Model Building (CNN)

↓

Model Training & Validation

↓

Performance Evaluation

↓

Result Visualization & Insights

↓

Deployment/Conclusion

3. Data Description

Dataset: **MNIST Handwritten Digits**

Samples: **70,000 images (60,000 train + 10,000 test)**

Format: **28x28 grayscale images**

Labels: **0 to 9 (digit classes)**

4. Data Processing

- ***Normalization of pixel values (0–255 to 0–1)***
- ***Reshaping images for model compatibility***
- ***One-hot encoding of labels***
- ***Splitting data into training, validation, and test sets***

5. Exploratory Data Analysis (EDA)

- ***Visualizing sample images from each digit class***
- ***Class distribution check***
- ***Plotting pixel intensity heatmaps***
- ***Analyzing image variance and data imbalance***

6. Feature Engineering

Although deep learning handles feature extraction, optional steps include:

- **Augmenting images (rotation, scaling, shifting)**
- **Flattening images (for non-CNN models)**
- **PCA (for dimensionality reduction in EDA)**

7. Model Building

Architecture: Convolutional Neural Network (CNN)

Input → Conv2D → ReLU → MaxPooling → Dropout → Dense → Softmax

Loss Function: Categorical Crossentropy

Optimizer: Adam

Metrics: Accuracy

8. Visualization of Results & Model Insights

- *Confusion matrix*

- *Accuracy & loss curves*

- *Incorrect predictions visualization*

- *Feature map visualizations (optional)*

9. *Tools and Technology Used*

- *Python*

- *Libraries: TensorFlow/Keras, NumPy, Pandas, Matplotlib, seaborn.*

- *Jupyter Notebook*