Introduction

# Purpose:

The purpose of this document is to outline the functional and non-functional requirements for **InkVision**, a **Handwritten Digit Recognition App** that identifies and classifies handwritten digits (0-9) from images using a deep learning model. The system will allow users to draw or upload digit images and receive real-time predictions.

# Scope:

InkVision will provide an interactive **React-based frontend** and a **Python (FastAPI) backend** integrated with a **CNN model trained on the MNIST dataset**. The application will process handwritten digit images and classify them using deep learning, with potential expansion into full Optical Character Recognition (OCR).

Overall Description

# Product Perspective:

InkVision is a **standalone** web application comprising:

* A **React frontend** for user interaction and input.
* A **FastAPI backend** for handling image processing and inference.
* A **CNN model (PyTorch)** trained on the MNIST dataset.

# Product Features:

* image upload.
* **Real-time classification** of handwritten digits (0-9).
* **Confidence scores** for model predictions.
* **Data preprocessing** (grayscale conversion, resizing).
* **REST API** for digit classification requests.
* **Future scope**: Expansion to full OCR.

# User Characteristics

* **Students & researchers** interested in AI/ML.
* **Developers** exploring computer vision applications.
* **General users** wanting real-time digit recognition.

# Operating Environment

* **Frontend:** React (runs on modern web browsers).
* **Backend:** Python (FastAPI, PyTorch).
* **Hosting:** Local server / Cloud deployment (AWS, Render, or Heroku).

Functional Requirements

1. **The system shall allow users to upload an image of a handwritten digit.**
2. **The system shall preprocess images (convert to grayscale, resize to 28x28 pixels).**
3. **The system shall classify the digit using a trained CNN model.**
4. **The system shall return the predicted digit with a confidence score.**
5. **The system shall display the classification result in real-time.**
6. **The system shall provide a REST API endpoint (/predict) for image classification requests.**

Non-Functional Requirements

1. **User Interface:** The system shall provide an intuitive and responsive UI for uploading images.
2. **Performance:** The system shall classify digits in under 2 seconds.
3. **Reliability:** The backend API shall maintain at least **99% uptime**.
4. **Security:** Uploaded images shall not be stored permanently, ensuring user privacy.
5. **Scalability:** The system shall support concurrent API requests.

System Requirements

# Hardware Requirements

* **Frontend:** Any device with a modern web browser.
* **Backend:** Server with a GPU (for training) or CPU (for inference).

# Software Requirements

* **Frontend:** React, Axios, TailwindCSS.
* **Backend:** Python, FastAPI, PyTorch, OpenCV.

Prototype Design

