Personal Project

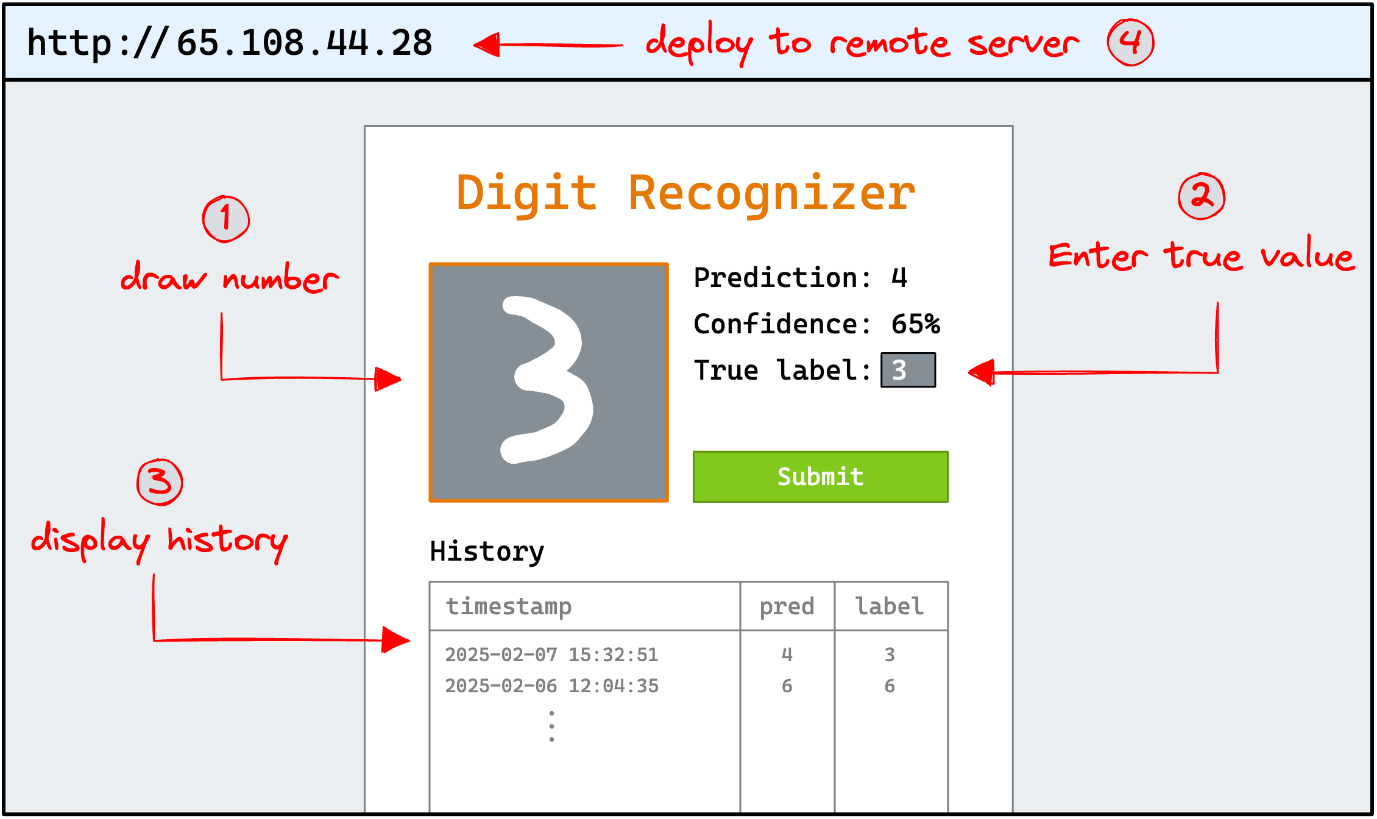
**Overview**

Building a small, end-to-end application on a self-managed server is one of the best ways to prepare for our programme. It gives you hands-on experience with the same tools and workflows (PyTorch, Docker, databases, and deployment) that we'll use throughout the course.

If you already have a clear idea for something you would like to build, feel free to pursue that. However, our recommended project is to build, containerize, and deploy an MNIST digit classifier.

**Recommended Project Brief**

Goal: Build, containerize, and deploy a simple digit-recogniser trained on the MNIST dataset.



1. Train a PyTorch Model
   * Develop a basic PyTorch model to classify handwritten digits from the MNIST dataset.
   * Train it locally and confirm that it achieves a reasonable accuracy.
2. Interactive Front-End
   * Create a web interface (using Streamlit) where users can draw a digit on a canvas or input area.
   * When the user submits the drawing, the web app should run the trained PyTorch model to produce:
     + Prediction: the model's guess at the digit (0–9).
     + Confidence: the model's probability for its prediction.
     + True Label: allow the user to manually input the correct digit so you can gather feedback.
3. Logging with PostgreSQL (MLI\_Project\_Work is the password)
   * Every time a prediction is made, log these details to a PostgreSQL database:
     + Timestamp
     + Predicted digit
     + User-provided true label
4. Containerization with Docker
   * Use Docker to containerize:
     + The PyTorch model/service
     + The Streamlit web app
     + The PostgreSQL database
   * Use Docker Compose to define your multi-container setup in a docker-compose.yml file.
5. Deployment
   * Set up a self-managed server (e.g., Hetzner's basic instance) or any other environment where you can install Docker and control the deployment end-to-end.
   * Deploy your containerized application to the server and make it accessible via a public IP or domain.
6. Add project to GitHub
   * Add your project to GitHub.
   * Make sure to include a README with a link to the live application.
   * Share the link to your GitHub repository with us via the application form.