

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Proposal of
Foundation Internship
Project Title: Smart Sport Shop

Instructor: Kim Ngoc Bach
Student Name : Bui Xuan Hai
Student ID : B22DCAT105
Lớp : E22CQCN05-B

Hà Nội - 2025

INTERNSHIP PROPOSAL

I. Project Introduction

This project aims to develop a web application that supports automatic recognition of vehicle license plates from uploaded images or videos. The system uses a deep learning model (trained by myself) to detect and recognize license plate numbers. This can be applied in real-life scenarios such as entry/exit management in parking lots or garages.

Users can interact with the system via a web interface, upload photos or videos, and the system will display the detected license plates. The backend is implemented using Flask/FastAPI, while the frontend is designed using HTML, CSS, and JavaScript (with optional integration of Bootstrap or Tailwind CSS).

I chose this topic because I am passionate about computer vision and want to become an AI engineer. This project allows me to apply AI models like YOLOv5 and PaddleOCR to a real-world task. It also helps me gain hands-on experience in building AI-powered web applications.

II. Key Features

- Allow users to upload vehicle images or videos.
- Automatically detect and recognize license plate numbers.
- Display recognition results in a user-friendly format.
- Possibility to store recognition history or export results.
- Designed for application in garage or parking management systems.

III. Technologies Used

- **Frontend:** HTML, CSS, JavaScript (Bootstrap/Tailwind CSS).
- **Backend:** Flask/FastAPI (Python).
- **AI Model:** YOLO-based license plate detection + OCR model (e.g., PaddleOCR or custom model).
- **Storage:** Local server
- **Label Studio:** Label datasets

I use **HTML, CSS, and JavaScript** (with Bootstrap or Tailwind CSS) for the frontend to create a responsive and user-friendly interface. For the backend, I choose **Flask or FastAPI** because they are lightweight, easy to integrate with AI models in Python, and well-suited for building RESTful APIs. The AI part uses **YOLO-based models** for accurate and real-time license plate detection, combined with **PaddleOCR** to extract text from plates effectively. These technologies work together to build a complete and efficient web-based AI application.

IV. Development Plan

Week 1-2:

- Finalize project idea and requirements.
- Research relevant technologies (YOLOv5, PaddleOCR, Flask/FastAPI, HTML/CSS/JS).
- Collects datasets about license plate and use Label Studio to labels on dataset

Week 3-4:

- Train or fine-tune YOLOv5 model for license plate detection.
- Test initial detection results.

- Integrate PaddleOCR and test OCR on sample images.

Week 5-6:

- Design basic frontend layout (HTML, CSS with Bootstrap or Tailwind CSS).
- Set up backend structure with Flask or FastAPI.
- Connect frontend to backend via API endpoints.
- Implement image upload feature from frontend to backend.

Week 7-8:

- Integrate YOLOv5 detection and PaddleOCR extraction into backend API.
- Return detection + recognition results to frontend.
- Build local database (SQLite/MySQL) to store uploaded images and extracted license plate numbers.(Optional)
- Save results into the database.(Optional)

Week 9-10:

- Conduct end-to-end testing: upload → detect → recognize → save to database(Optional) → display results.
- Optimize model inference and API speed.
- Final testing and bug fixing.
- Complete documentation and prepare for presentation.

