Installation and Setup Guide for ASL TRANSLATION SYSTEM

**1. Prerequisites**

1. **Python 3.8+** installed and on your PATH

Verify with following Code by running in terminal:

Code:

python --version

pip –version

1. **MySQL** (or another supported RDBMS) running locally or reachable remotely
2. **Git** (to clone the repo)

**Git hub soure repo link:** <https://github.com/bsameer12/Production_Project_Final_Year/tree/main/%20Implementation/WebApp%20Source%20Code/ASl_UI>

**Or download zip file from link below:**

<https://drive.google.com/file/d/1pCOGzDwd6ERGhvzvdhfEeTVZpK-PQvmk/view?usp=share_link>

1. **Virtualenv** (optional but recommended)
2. **An SMTP-capable email account** (Gmail App Password, SendGrid, etc.)
3. **Google Gemini API Key** (for English sentence generation)

**2. Clone & Virtual Environment**

**Run the following command in terminal:**

**CODE:**

**# 1. Clone your repo**

**git clone <your-repo-url>**

**cd <project\_root>**

**# 2. Create & activate a venv**

**python -m venv venv**

**# macOS/Linux:**

**source venv/bin/activate**

**# Windows (PowerShell):**

**.\venv\Scripts\Activate.ps1**

**# 3. Upgrade pip**

**pip install --upgrade pip**

**3. Install Python Dependencies**

**Run the following command in terminal:**

**CODE:**

**pip install -r requirements.txt**

Your requirements.txt should include at least:

* Django>=4.0
* mysqlclient (if using MySQL)
* django-environ or python-dotenv
* django-cors-headers
* any Gemini client library you use

**4. Environment Variables**

Create a file named .env in your project root (next to manage.py) with:

CODE:

**# Django**

**SECRET\_KEY='your-django-secret-key'**

**# Database (MySQL example)**

**DB\_NAME='your\_db\_name'**

**DB\_USER='your\_db\_user'**

**DB\_PASSWORD='your\_db\_password'**

**DB\_HOST='localhost'**

**DB\_PORT='3306'**

**# Email (Gmail example)**

**EMAIL\_HOST='smtp.gmail.com'**

**EMAIL\_PORT=587**

**EMAIL\_HOST\_USER='your.email@gmail.com'**

**EMAIL\_HOST\_PASSWORD='your-app-password'**

**EMAIL\_USE\_TLS=True**

**# Gemini**

**GEMINI\_API\_KEY='your-gemini-api-key'**

**Note:** In settings.py you should have something like:

**import os**

**import environ**

**BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent**

**env = environ.Env()**

**environ.Env.read\_env(os.path.join(BASE\_DIR, '.env'))**

**SECRET\_KEY = env('SECRET\_KEY')**

**DATABASES = {**

**'default': {**

**'ENGINE': 'django.db.backends.mysql',**

**'NAME': env('DB\_NAME'),**

**'USER': env('DB\_USER'),**

**'PASSWORD': env('DB\_PASSWORD'),**

**'HOST': env('DB\_HOST'),**

**'PORT': env('DB\_PORT'),**

**}**

**}**

**EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'**

**EMAIL\_HOST = env('EMAIL\_HOST')**

**EMAIL\_PORT = env('EMAIL\_PORT')**

**EMAIL\_HOST\_USER = env('EMAIL\_HOST\_USER')**

**EMAIL\_HOST\_PASSWORD = env('EMAIL\_HOST\_PASSWORD')**

**EMAIL\_USE\_TLS = env.bool('EMAIL\_USE\_TLS', default=True)**

**GEMINI\_API\_KEY = env('GEMINI\_API\_KEY')**

**INSTALLED\_APPS += [**

**'accounts',**

**'asl',**

**'corsheaders',**

**]**

**MIDDLEWARE = [**

**'corsheaders.middleware.CorsMiddleware',**

**# … other middleware …**

**]**

**CORS\_ALLOW\_ALL\_ORIGINS = True**

**STATIC\_URL = '/static/'**

**STATICFILES\_DIRS = [BASE\_DIR / 'static']**

**MEDIA\_URL = '/media/'**

**MEDIA\_ROOT = BASE\_DIR / 'media'**

**LOGIN\_URL = '/accounts/login/'**

**LOGIN\_REDIRECT\_URL = '/predict/'**

**LOGOUT\_REDIRECT\_URL = '/accounts/login/'**

**5. Database Setup**

**1. Create the database** in MySQL:

**SQL CODE:**

**CREATE DATABASE your\_db\_name CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;**

**GRANT ALL ON your\_db\_name.\* TO 'your\_db\_user'@'localhost' IDENTIFIED BY 'your\_db\_password';**

**2. Apply migrations**:

**Run the following command in terminal:**

**CODE:**

**python manage.py migrate**

**3. Create a superuser**:

**Run the following command in terminal:**

**CODE:**

**python manage.py createsuperuser**

**6. Static & Media Files**

**1. Collect static** (for production):

**Run the following command in terminal:**

**CODE:**

**python manage.py collectstatic**

2. **Ensure media folder exists**:

**Run the following command in terminal:**

**CODE:**

**mkdir -p media**

**7. Background Cleanup Task**

This project includes an automatic video-cleanup task. To enable it:

If you’ve implemented it as a custom management command (e.g. cleanup\_videos), schedule it via **cron**:

**Run the following command in terminal:**

**CODE:**

0 3 \* \* \* cd /path/to/project && /path/to/venv/bin/python manage.py cleanup\_videos

Or, if you’re using **django-crontab**, add to settings.py:

**CODE:**

**CRONJOBS = [**

**('0 3 \* \* \*', 'django.core.management.call\_command', ['cleanup\_videos']),**

**]**

then run:

**Run the following command in terminal:**

**CODE:**

**python manage.py crontab add**

(Adjust the schedule to your needs.)

**8. Run the Development Server**

**Run the following command in terminal:**

**CODE:**

**// To activate Virtual Enivronment if running in virtual environment:**

**source venv/bin/activate**

**// To run server:**

**python manage.py runserver**

Open your browser at <http://localhost:8000/> and you should see the login page.

**9. Verifying Key Features**

1. **Registration & Email Verification** – sign up and check that an email with a verification link arrives.

2. **ASL Prediction** – navigate to /predict/ and test your webcam or landmark‐upload interface.

3. **History & Audit Logs** – visit your profile/history pages to confirm predictions, logs, and generated videos/sentences are stored.

4. **Admin Site** – log in as superuser at /admin/ to view/manage users, logs, sentences, videos.

**Troubleshooting Tips:**

* **Database errors**: verify your .env variables and that MySQL is running / accessible.
* **Email issues**: ensure your SMTP credentials and “Allow less secure apps” (or App Password) are properly configured.
* **Gemini API**: test a small script to confirm your GEMINI\_API\_KEY works before running the full app.

That’s it! Your “ASL Sign Translator” should now be up and running locally. If you deploy to production, remember to turn off DEBUG, configure ALLOWED\_HOSTS, secure your SECRET\_KEY, and serve static/media files via a proper web server.