

## Predict\_Xplore – Setup Guide

Predict\_Xplore consists of a **Django backend** and a **React (Vite) frontend**. Follow the steps below to set up the project locally.

---

### Prerequisites

Ensure the following are installed on your system:

- **Python**  $\geq 3.10$
  - **Node.js**  $\geq 16$
  - **npm** or **yarn**
  - **Git**
  - **Docker Desktop** (required)
  - **PostgreSQL** (used in production; optional for local setup)
  - **virtualenv** (recommended)
- 

### 1 Clone the Repository

```
git clone https://github.com/Predict-Xplore-IIT-B/Predict-Xplore.git
```

```
cd Predict-Xplore
```

### Repository Structure

```
Predict-Xplore/
```

```
|
```

```
|— backend/
```

```
|  └─ Xplore/      # Django backend
```

```
|
```

```
└─ frontend/      # React frontend (Vite)
```

---

### 2 Backend Setup (Django)

#### Step 1: Navigate to Backend

```
cd backend/Xplore
```

#### Step 2: Create & Activate Virtual Environment

```
python -m venv venv
```

**Activate it:**

- macOS / Linux:  
source venv/bin/activate
- Windows:  
venv\Scripts\activate

### Step 3: Install Dependencies

```
pip install -r requirements.txt
```

If requirements.txt is missing or incomplete, install manually:

```
pip install django djangorestframework social-auth-app-django \
python-decouple google-api-python-client google-auth-http2 \
google-auth-oauthlib
```

### Step 4: Grad-CAM Dependency Check

Ensure this **local wheel file** exists:

```
backend/Xplore/libs/grad_cam-1.5.5-py3-none-any.whl
```

If missing, add it before proceeding.

---

### Step 5: Generate Django Secret Key

```
python -c "from django.core.management.utils import get_random_secret_key;
print('DJANGO_SECRET_KEY=' + get_random_secret_key())"
```

**OR**

```
python generate_secret.py
```

---

### Step 6: Create .env File

Create a .env file inside backend/Xplore/:

```
EMAIL_ID=""
```

```
EMAIL_PASS=""
```

```
GOOGLE_OAUTH_CLIENT_ID=
```

```
GOOGLE_OAUTH_CLIENT_SECRET=
```

```
DJANGO_SECRET_KEY=""
```

 **Important:**

Fill all required values **before starting the server**.

---

### Step 7: Apply Migrations

```
python manage.py makemigrations
```

```
python manage.py migrate
```

### **Step 8: Create Superuser (Optional)**

```
python manage.py createsuperuser
```

### **Step 9: Run Backend Server**

```
python manage.py runserver
```

Backend runs at:

<http://localhost:8000/>

---

## **3 Frontend Setup (React + Vite)**

### **Step 1: Open a New Terminal**

```
cd frontend
```

### **Step 2: Install Dependencies**

```
npm install
```

### **Step 3: Start Frontend Server**

```
npm run dev
```

Frontend runs at:

<http://localhost:5173/>

---

## **Google OAuth Setup**

### **Step 1: Create Google Cloud Project**

- Visit **Google Cloud Console**
  - Create a new project (e.g., *Predict Xplore*)
- 

### **Step 2: Configure OAuth Consent Screen**

- Go to **APIs & Services → OAuth consent screen**
- Choose **External**
- Fill:
  - App name
  - Support email
  - Developer contact info

- Save
- 

### Step 3: Create OAuth Client ID

- Go to **APIs & Services** → **Credentials**
- Click **Create Credentials** → **OAuth Client ID**
- Select:
  - **Application type:** Web application
  - **Name:** PredictXplore

### Authorized JavaScript Origins

`http://localhost:8000`

`http://localhost:5173`

### Authorized Redirect URI

`http://localhost:8000/auth/complete/google-oauth2/`

 Changes may take **5 minutes to a few hours** to reflect.

---

### Step 4: Configure Backend Credentials

- Copy values into `.env`:

`GOOGLE_OAUTH_CLIENT_ID=your_client_id`

`GOOGLE_OAUTH_CLIENT_SECRET=your_client_secret`

- Download `credentials.json`
- Paste its content into:

`backend/util/credentials.json`

---

## 5 Gmail App Password Setup (For Email / OTP)

### Step 1: Enable 2-Step Verification

- Go to **myaccount.google.com/security**
- Enable **2-Step Verification**

### Step 2: Generate App Password

- Go to **Security** → **App passwords**
- Select:
  - App: Mail

- Device: Other (e.g., DjangoApp)
- Generate password (16 characters)

### Step 3: Update .env

EMAIL\_ID="yourgmail@gmail.com"

EMAIL\_PASS="abcdefghijklmnop"

#### Notes

- Never use your real Gmail password
  - Regenerate app password if Gmail password changes
- 

### 6 Test the Application

- Frontend:
  - <http://localhost:5173/>
  - Try **Login / OTP / Google Auth**
  - Django Admin:
  - <http://localhost:8000/admin>
- 

### 7 Troubleshooting

Issue	Solution
redirect_uri_mismatch	Ensure exact redirect URI is added in Google Console & credentials.json
invalid_grant (Gmail API)	Delete token.json and re-authenticate
ModuleNotFoundError	Activate virtual environment
React API errors	Check Django CORS configuration
Reset database	python manage.py flush

---

### Docker Requirement

- **Docker Desktop must be installed and running**
- Required for containerized services used in the project