Procedure To Run Project

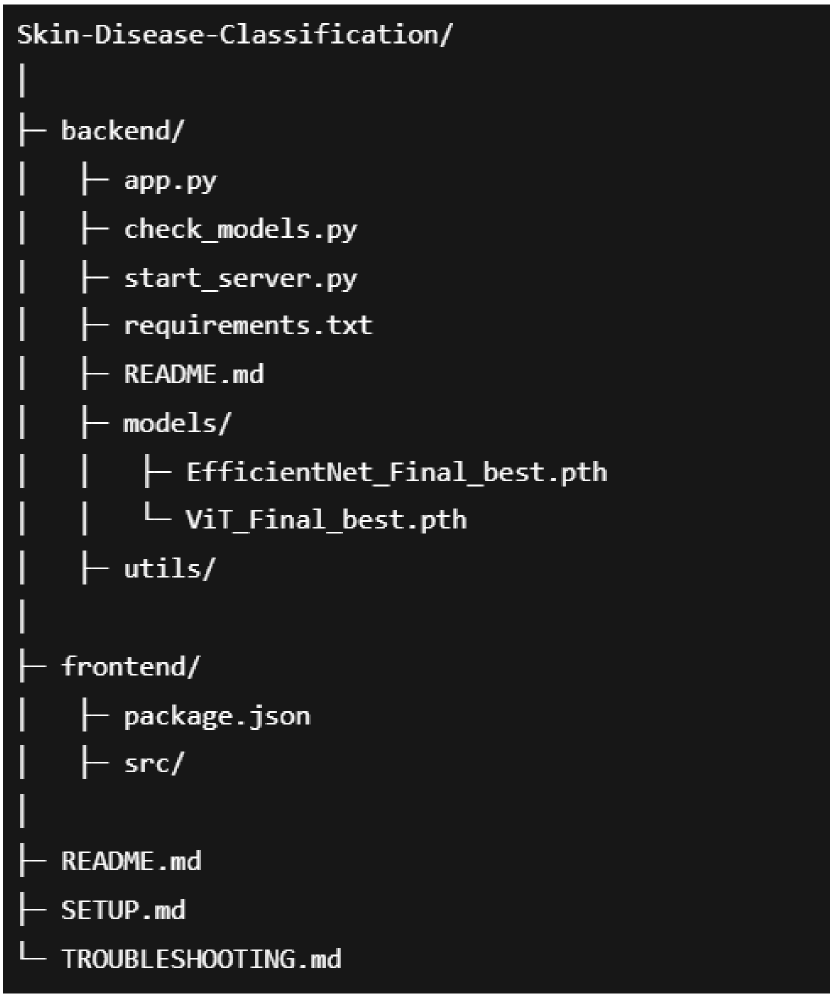
1. Prerequisites

Install the following before running the project:

1. **Python** 3.9+
2. **Node.js** (with npm) 16+
3. **Git** (optional, if cloning from repo)
4. Basic hardware:
    - CPU: i5 / Ryzen or better
    - RAM: ≥ 8 GB

If you have a GPU with CUDA, you can install the GPU version of PyTorch, otherwise CPU-only is fine (slower but works).

2. Project Directory Structure



```
Skin-Disease-Classification/
│
├─ backend/
│   ├─ app.py
│   ├─ check_models.py
│   ├─ start_server.py
│   ├─ requirements.txt
│   ├─ README.md
│   ├─ models/
│   │   ├─ EfficientNet_Final_best.pth
│   │   └─ ViT_Final_best.pth
│   ├─ utils/
│
├─ frontend/
│   ├─ package.json
│   ├─ src/
│
├─ README.md
├─ SETUP.md
└─ TROUBLESHOOTING.md
```

3. Backend Setup (Flask + AI Ensemble Inference)

Step 1 — Open Terminal and navigate to backend

```
cd backend
```

Step 2 — Create & activate Virtual Environment

```
# Windows CMD / PowerShell
python -m venv venv
venv\Scripts\activate

# Linux / macOS
python3 -m venv venv
source venv/bin/activate
```

Step 3 — Install dependencies

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

Step 4 — Verify model paths

Ensure `.pth` model files exist inside:

```
backend/models/
```

Model loading inside `app.py` or `check_models.py` should reference:

```
./models/EfficientNet_Final_best.pth
./models/ViT_Final_best.pth
```

Step 5 — Test model loading (optional recommended)

```
python check_models.py
```

Step 6 — Run Backend Server

```
python start_server.py
```

or if directly using app.py:

```
python app.py
```

Backend will start at

```
http://localhost:5000
```

---

4. Frontend Setup (React Web UI)

Step 1 — Open a new terminal

```
cd frontend
```

Step 2 — Install dependencies

```
npm install
# or
yarn
```

Step 3 — Configure API endpoint

Inside `frontend/src/api.js` or `frontend/src/api.ts`:

```
export const API_URL = "http://localhost:5000";
```

Step 4 — Run frontend

```
npm start
# or
yarn start
```

Frontend runs at:

```
http://localhost:3000
```

---

5. Running the Application End-to-End

1. Start **backend** in terminal:
2. `python start_server.py`
3. Start **frontend** in another terminal:
4. `npm start`
5. Open web app in browser:
6. `http://localhost:3000`
7. Upload or capture a skin image.
8. Click **Analyze / Predict**
9. System returns:
    - Benign / Malignant classification
    - Confidence %
    - Probability chart
    - **Grad-CAM heatmap**
    - Preventive measures & explanation

---

6. Common Errors & Solutions

Issue	Possible Fix
Model not found	Check model file paths inside `backend/models/`
CORS error	Add `from flask_cors import CORS; CORS(app)`
Frontend cannot connect to backend	Confirm correct API URL

Issue	Possible Fix
Pip errors installing torch	Install CPU/GPU-specific PyTorch build
“Port already in use”	Change 5000 or close other running servers

7. Quick Commands Summary

```
# BACKEND
cd backend
python -m venv venv
venv\Scripts\activate
pip install -r requirements.txt
python start_server.py

# FRONTEND (New Terminal)
cd frontend
npm install
npm start
```

8. Completion Output Expected

After successful execution:

- Backend running at `localhost:5000`
- Frontend running at `localhost:3000`
- Images classify with **real-time results & heatmap**