**🔹 Step 1: Download the Project**

* Download the Project Scaffold.
* You will receive a zip file with the project structure, or you can clone/download from a version control system.

**🔹 Step 2: Open in VS Code**

* Open VS Code.
* Go to **File → Open Folder** and select the extracted project folder.

**🔹 Step 3: Setup Python Environment**

1. Open the **VS Code Terminal** (Ctrl + backtick).
2. Create a virtual environment:
3. python -m venv venv
4. Activate it:
   * **Windows**:
   * venv\Scripts\activate
   * **Linux/Mac**:
   * source venv/bin/activate
5. Install dependencies:
6. pip install -r requirements.txt

**🔹 Step 4: Prepare Data**

Run preprocessing (downloads dataset from Hugging Face and saves artifacts):

python data\_prep.py

This will:

* Download dataset (Tobi-Bueck/customer-support-tickets).
* Clean, tokenize, pad sequences.
* Save artifacts into artifacts/.

**🔹 Step 5: Train the Model**

Train the Many-to-One LSTM:

python train.py --epochs 6 --batch\_size 64

This will:

* Train the RNN model.
* Save the best model to saved\_models/.
* Save confusion matrix to confusion\_matrix.png.

**🔹 Step 6: Test Inference + Gemini Reply**

Try inference on a sample ticket:

python infer\_and\_reply.py

If you haven’t set a Gemini API key, you’ll see a **fallback polite reply**.  
If you have a Gemini API key:

* Create a .env file in project root:
* GEMINI\_API\_KEY=AIzaSyAJEm7Hz9S5UFakFpvoGAXcGNTi8exnuFQ
* Then rerun the script.

**🔹 Step 7: Run Streamlit App (UI)**

Launch the UI to test interactively:

streamlit run app\_streamlit.py

This will open a web page (<http://localhost:8501>) where you can:

* Paste a customer ticket text.
* See the predicted queue.
* View a generated reply.

✅ **End-to-end flow:**  
Dataset → Preprocess → Train → Evaluate → Predict + Auto-reply → Web UI

Do you want me to **regenerate the zip with the full scaffold code (instead of placeholders)** so you can just unzip and run without needing to copy the full code manually?