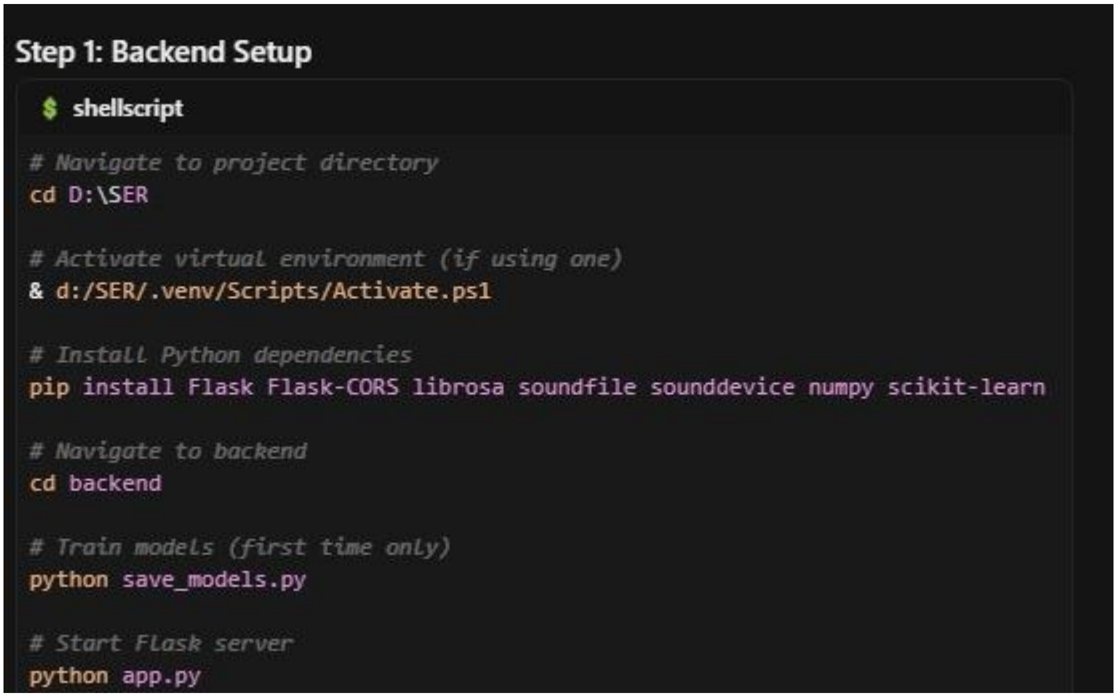
Backend setup



Step 1: Backend Setup

$ shellscript

```
# Navigate to project directory
cd D:\SER

# Activate virtual environment (if using one)
& d:/SER/.venv/Scripts/Activate.ps1

# Install Python dependencies
pip install Flask Flask-CORS librosa soundfile sounddevice numpy scikit-learn

# Navigate to backend
cd backend

# Train models (first time only)
python save_models.py

# Start Flask server
python app.py
```

Navigate to project directory

cd D:\SER

& d:/SER/.venv/Scripts/Activate.ps1

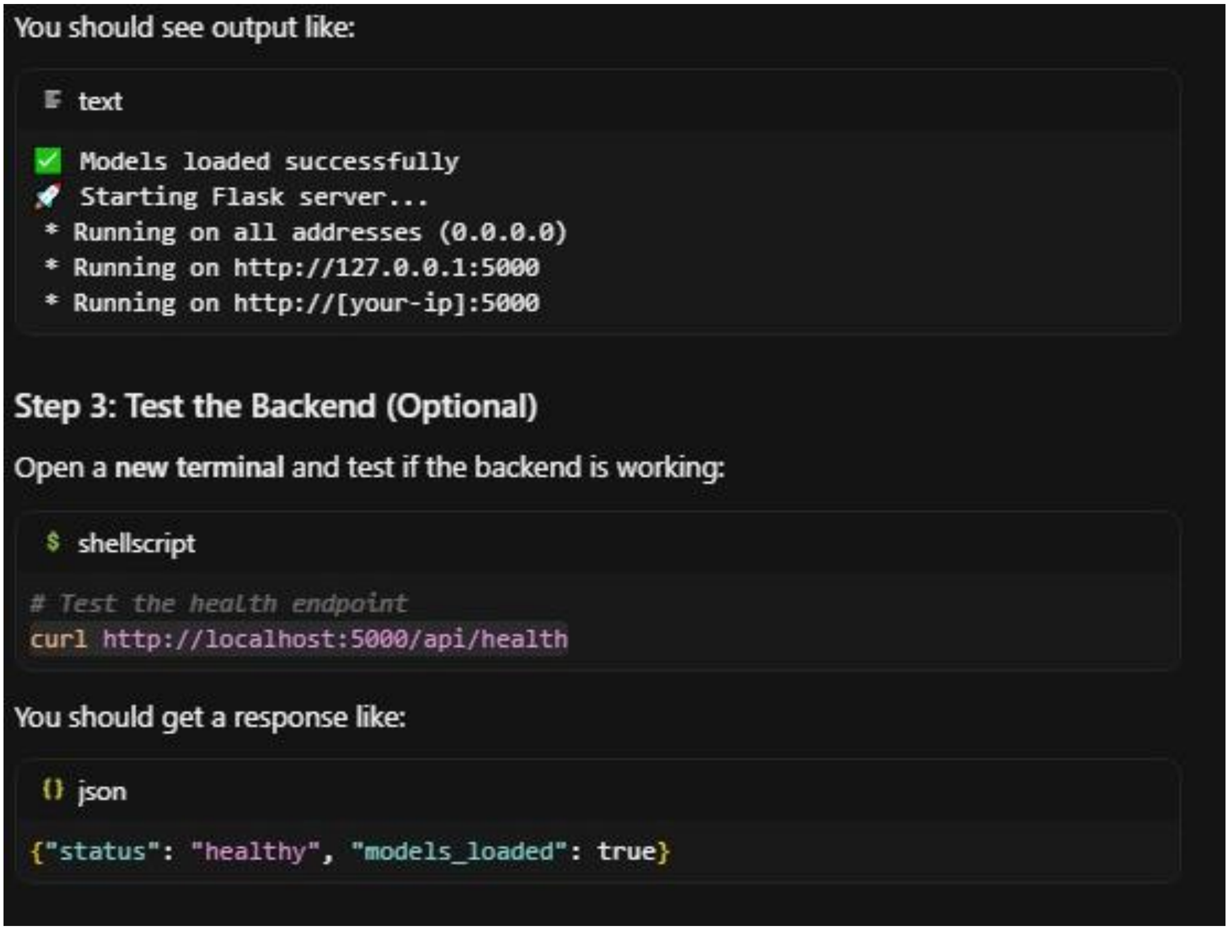
pip install Flask Flask-CORS librosa soundfile sounddevice numpy scikit-learn lightgbm xgboost imbalanced-learn matplotlib seaborn pyaudio

Navigate to backend

cd backend

python save_models.py

python app.py



You should see output like:

```text
✅ Models loaded successfully
🚀 Starting Flask server...
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://[your-ip]:5000
```

Step 3: Test the Backend (Optional)

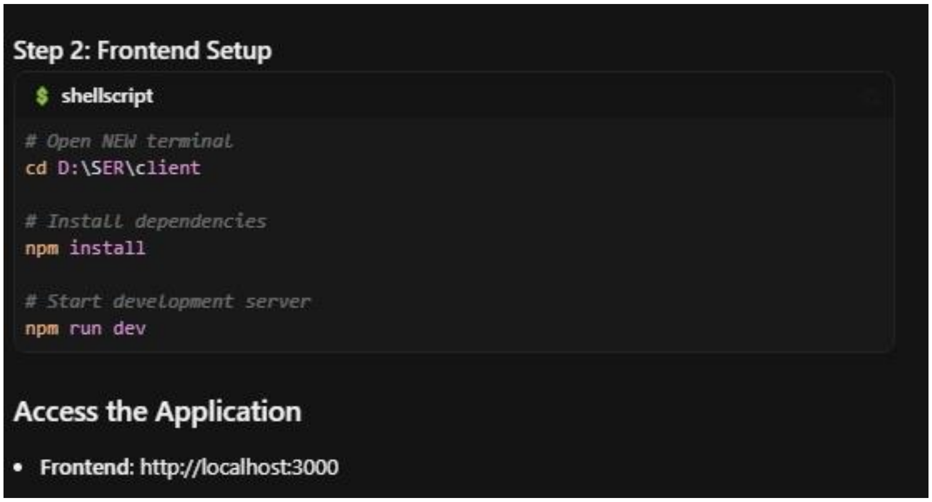
Open a **new terminal** and test if the backend is working:

```shellscript
# Test the health endpoint
curl http://localhost:5000/api/health
```

You should get a response like:

```json
{"status": "healthy", "models_loaded": true}
```

Frontend setup



Step 2: Frontend Setup

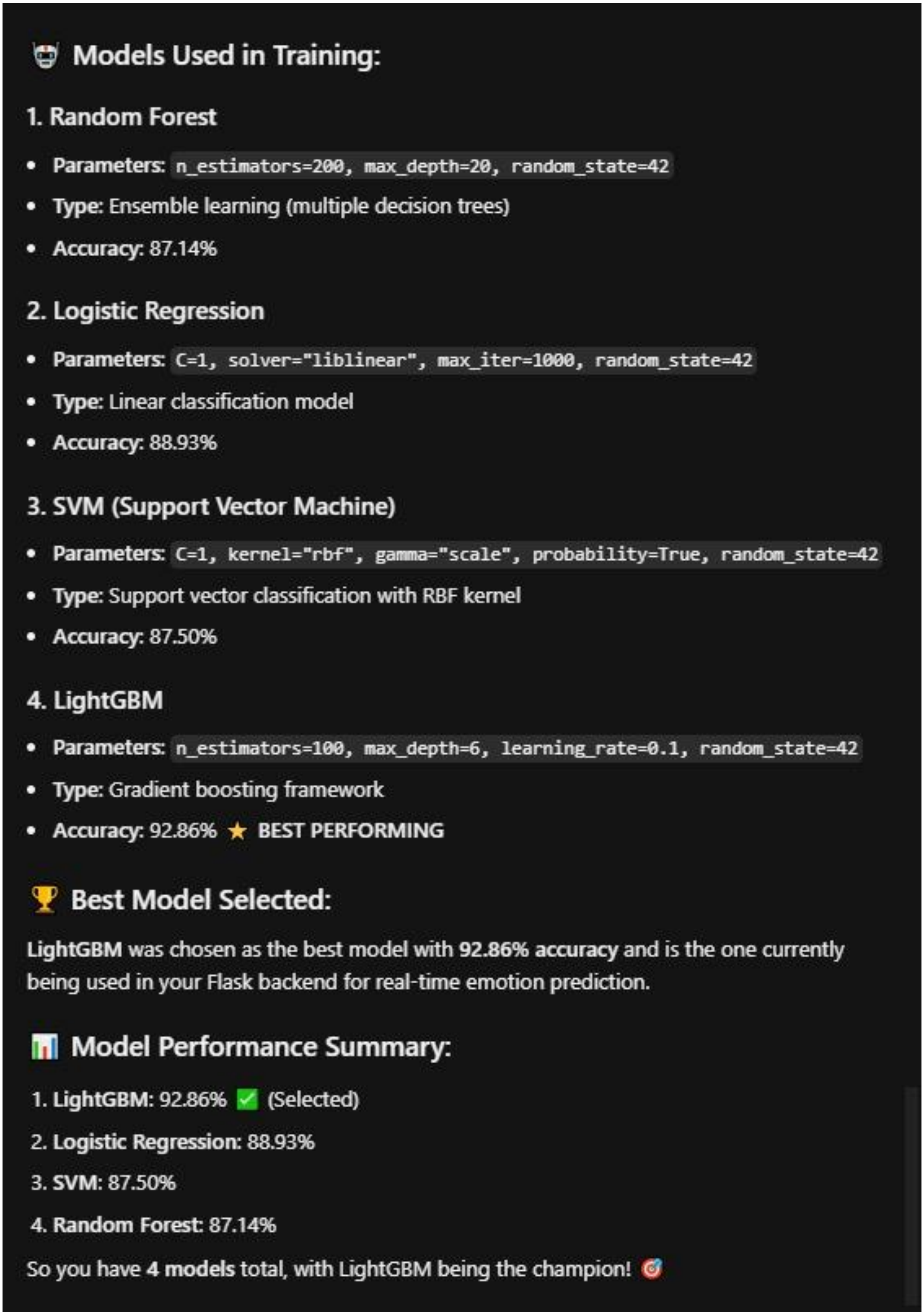
```shellscript
# Open NEW terminal
cd D:\SER\client

# Install dependencies
npm install

# Start development server
npm run dev
```

Access the Application

- **Frontend:** http://localhost:3000



🤖 Models Used in Training:

1. Random Forest

- **Parameters:** `n_estimators=200, max_depth=20, random_state=42`
- **Type:** Ensemble learning (multiple decision trees)
- **Accuracy:** 87.14%

2. Logistic Regression

- **Parameters:** `C=1, solver="liblinear", max_iter=1000, random_state=42`
- **Type:** Linear classification model
- **Accuracy:** 88.93%

3. SVM (Support Vector Machine)

- **Parameters:** `C=1, kernel="rbf", gamma="scale", probability=True, random_state=42`
- **Type:** Support vector classification with RBF kernel
- **Accuracy:** 87.50%

4. LightGBM

- **Parameters:** `n_estimators=100, max_depth=6, learning_rate=0.1, random_state=42`
- **Type:** Gradient boosting framework
- **Accuracy:** 92.86% ⭐ **BEST PERFORMING**

🏆 Best Model Selected:

LightGBM was chosen as the best model with **92.86% accuracy** and is the one currently being used in your Flask backend for real-time emotion prediction.

📊 Model Performance Summary:

1. **LightGBM:** 92.86% ✅ (Selected)
2. **Logistic Regression:** 88.93%
3. **SVM:** 87.50%
4. **Random Forest:** 87.14%

So you have **4 models** total, with LightGBM being the champion! 🎯