# 🎯 NETRAX AI - Complete Integration Guide

## 📁 FINAL PROJECT STRUCTURE

```
netrax_ai/
├── frontend/
│   └── index.html              # Your UI (provided)
│
├── modules/
│   └── body_detection/
│       ├── __init__.py
│       ├── camera.py
│       ├── pose.py
│       ├── gesture.py
│       ├── tracking.py
│       ├── adapter.py
│       └── body_detection.py
│
├── config/
│   ├── body_detection_config.json
│   └── gesture_mappings.json
│
├── backend_server.py           # MAIN SERVER (start this)
└── requirements.txt
```

---

## 🚀 STEP-BY-STEP SETUP

### Step 1: Install Dependencies

```bash
pip install fastapi uvicorn opencv-python mediapipe numpy websockets psutil
```

### Step 2: File Placement

1. **Save your UI** as `frontend/index.html`
2. **All Python files** you provided go in their locations (already provided)
3. **Config files** already in `config/` folder

**Step 3: Start the Backend Server**

```
bash

python backend_server.py
```

**Expected Output:**

```
INFO:     Started server process
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000
[Camera] Opened camera 2: 640x480 @ 30fps
```

**Step 4: Open the Frontend**

Open `frontend/index.html` in your browser (Chrome/Edge recommended)

**Or serve it locally:**

```
bash

# In frontend folder
python -m http.server 8080
```

Then visit: `http://localhost:8080`

---

## ✅ VERIFICATION CHECKLIST

**When you open the UI:**

**✅ Initial Load (First 3 seconds):**

- Black screen with red particle animation
- Red thread curtain animation opens
- "NETRAX AI" branding appears
- Closed eye appears in center
- Custom red cursor follows mouse

**✅ Click the Eye:**

- Eye opens smoothly
- UI panels fade in:

    - Video panel (top right)
    - Stats panel (bottom left)
    - Gestures panel (bottom right)
- "Click eye to close" message appears

✅ **Click "START DETECTION" button:**

- Eye closes for 3 seconds
- Eye reopens with RED GLOWING iris
- Video feed appears in top right (LIVE indicator)
- 100 small eyes spawn and spread
- Debug panel shows connection status
- Stats update in real-time

✅ **Backend Connected:**

- Debug panel shows: "✅ WebSocket CONNECTED!"
- Video feed shows camera stream
- FPS counter updates
- Gesture count increases when you make gestures
- Accuracy percentage shows

✅ **Make Gestures:**

- Peace sign ✌️ → Gesture box flashes
- Thumbs up 👍 → Volume up box flashes
- Stop ✋ → Pause box flashes
- Fist ✊ → Mute box flashes

---

## 🔧 TROUBLESHOOTING

**Problem: "WebSocket ERROR" in debug panel**

**Solution:**

```bash
```

```
# Check if backend is running
# Terminal should show:
INFO:     Uvicorn running on http://0.0.0.0:8000

# If not, start it:
python backend_server.py
```

**Problem: Camera not opening**

**Solution 1 - Find your camera ID:**

```
bash

python modules/body_detection/camera_test.py
```

**Solution 2 - Update config:** Edit `config/body_detection_config.json`:

```
json

{
  "camera_id": 0,  ⟵ Change this (try 0, 1, 2)
  ...
}
```

**Problem: Video feed not showing**

**Check:**

1. Backend server is running (`python backend_server.py`)
2. Camera permissions granted
3. Visit `http://localhost:8000/video_feed` directly - should show video

**Problem: Gestures not detected**

**Improve detection:**

1. Stand 1-2 meters from camera
2. Good lighting on face/hands
3. Plain background
4. Hold gestures for 0.5 seconds

**Lower sensitivity in config:**

```
json
```

```
{
  "gesture_min_confidence": 0.5,  ⟵ Lower value = easier detection
  "gesture_hold_time": 0.2,       ⟵ Faster recognition
  ...
}
```

---

## 🎮 USAGE FLOW

**Complete User Journey:**

1. **Open index.html** → Thread animation plays
2. **Eye appears** → Click eye to wake it
3. **Panels appear** → Click "START DETECTION"
4. **Eye transforms** → Closes, reopens red
5. **Small eyes spawn** → 100 eyes spread across screen
6. **Backend connects** → Video feed starts
7. **Make gestures** → System detects and responds
8. **Click main eye** → System closes with animation

---

## 📊 WHAT EACH FILE DOES

| File | Purpose |
|---|---|
| `frontend/index.html` | Your beautiful UI with animations |
| `backend_server.py` | FastAPI server for WebSocket + video streaming |
| `body_detection.py` | Main detection system orchestrator |
| `camera.py` | Webcam capture with threading |
| `pose.py` | MediaPipe pose/hand detection |
| `gesture.py` | Converts poses to gestures |
| `tracking.py` | Smooths jittery movements |
| `adapter.py` | Maps gestures to commands |

## 🔌 API ENDPOINTS

Your backend provides:

| Endpoint | Type | Purpose |
|---|---|---|
| http://localhost:8000/ | HTTP | Info page |
| http://localhost:8000/video_feed | HTTP Stream | MJPEG video |
| ws://localhost:8000/ws | WebSocket | Real-time stats + gestures |

### WebSocket Message Format:

### Stats Update (every 0.5s):

```json
{
  "type": "stats",
  "timestamp": 1234567890.123,
  "stats": {
    "fps": 28.5,
    "gesture_count": 5,
    "detection_count": 150,
    "confidence": 0.92
  },
  "cpu": 35.2
}
```

### Gesture Detected:

```json
{
  "type": "gesture_command",
  "timestamp": 1234567890.123,
  "command": "peace",
  "parameters": {
    "gesture_confidence": 0.95,
    "gesture_hand": "right"
  }
}
```

## 🎨 CUSTOMIZATION

### Change Gesture Mappings:

Edit `config/gesture_mappings.json`:

```
json

{
  "peace": {
    "action": "screenshot",
    "parameters": {}
  },
  "thumbs_up": {
    "action": "volume_up",
    "parameters": {"amount": 10}
  }
}
```

### Adjust Detection Sensitivity:

Edit `config/body_detection_config.json`:

```
json

{
  "pose_confidence": 0.5,          ⟵ Lower = easier detection
  "gesture_min_confidence": 0.6,   ⟵ Lower = easier recognition
  "gesture_hold_time": 0.3,        ⟵ Lower = faster response
  "pose_model_complexity": 1       ⟵ 0=Fast, 1=Balanced, 2=Accurate
}
```

### Change UI Colors:

Edit `frontend/index.html` CSS variables (around line 15-20):

```
css

/* Red theme (current) */
#ff0000, #cc0000, #8b0000

/* Blue theme */
#0066ff, #0044cc, #002288

/* Green theme */
#00ff00, #00cc00, #008800
```

---

## 🐛 DEBUG MODE

The UI has a built-in debug panel (bottom right) showing:

- 🚀 System initialization
- 📡 Connection attempts
- ✅ WebSocket status
- 📹 Video feed status
- 📊 Stats updates
- 👋 Gesture detections
- ❌ Errors

**To hide it**, edit `index.html` line ~690:

```javascript
const DEBUG = false;  // Change to false
```

---

## 🚦 STARTUP SEQUENCE

**Terminal (Backend):**

```
1. [Camera] Opened camera 2: 640x480 @ 30fps
2. [Pose] MediaPipe Holistic initialized
3. [Adapter] Initialized in callback mode
4. [BodyDetection] System started
5. INFO: Uvicorn running on http://0.0.0.0:8000
```

**Browser (Frontend):**

```
1. 🚀 NETRAX AI - Vision System Initialized
2. 📡 Attempting to connect to backend...
3. 🔗 Connecting to WebSocket...
4. ✅ WebSocket CONNECTED!
5. 📹 Video feed loaded
6. 👁️ Eye awakened
7. 👀 Small eyes spawned
8. 📊 Stats: FPS=28, Gestures=0
```

---

## ⚡ PERFORMANCE TIPS

### Optimize for Speed:

```json
{
  "camera_width": 640,
  "camera_height": 480,
  "pose_model_complexity": 0,
  "skip_frames": 1,
  "enable_smoothing": false
}
```

### Optimize for Accuracy:

```json
{
  "camera_width": 1280,
  "camera_height": 720,
  "pose_model_complexity": 2,
  "skip_frames": 0,
  "enable_smoothing": true
}
```

---

## 📝 QUICK START COMMANDS

```bash
```

```
# 1. Install everything
pip install fastapi uvicorn opencv-python mediapipe numpy websockets psutil

# 2. Test camera
python modules/body_detection/camera_test.py

# 3. Start backend
python backend_server.py

# 4. Open frontend
# Double-click frontend/index.html
# OR
cd frontend && python -m http.server 8080

# 5. Make gestures!
# Peace ✌️, Thumbs up 👍, Stop ✋, Fist ✊
```

---

## ✨ FEATURES WORKING

- ✅ Thread curtain opening animation
- ✅ Eye appears naturally after animation
- ✅ Click to wake → Eye opens smoothly
- ✅ Start detection → Eye closes & transforms
- ✅ Red glowing awakened eye with pulse
- ✅ 100 small eyes spawn from center
- ✅ Main eye tracks mouse cursor
- ✅ Small eyes look randomly
- ✅ Real-time video feed from camera
- ✅ WebSocket live updates
- ✅ Gesture detection with visual feedback
- ✅ Stats panel (FPS, gestures, accuracy)
- ✅ Glassmorphism UI panels
- ✅ Click eye to close with shrink animation
- ✅ Thread closing animation
- ✅ Debug panel for troubleshooting

---

## 🎯 FINAL CHECKLIST

- ☐ Backend running (`python backend_server.py`)
- ☐ Frontend open (`frontend/index.html`)
- ☐ Camera permissions granted

- [ ] WebSocket connected (check debug panel)
- [ ] Video feed showing
- [ ] Gestures being detected
- [ ] All animations smooth

---

## 🎉 YOU'RE DONE!

Your NETRAX AI vision system is now fully operational!

**Test it:**

1. Open `frontend/index.html`
2. Watch the thread animation
3. Click the eye
4. Click "START DETECTION"
5. Make gestures and watch them get detected!

**Enjoy your mystical AI vision system!** 👁️🔴