Quick Start Guide - Frontend

One-Minute Setup

1. Install Dependencies

```
cd frontend
npm install
```

2. Start Backend

```
cd backend
pip install -r requirements.txt
uvicorn main:app --reload --port 8000
```

Backend will run on `http://localhost:8000`

3. Start Frontend

```
cd frontend
npm run dev
```

Frontend will run on `http://localhost:5173`

4. Open Browser

Go to `http://localhost:5173`

First Steps in the App

Option A: Test with Image Upload

1. Click **"📤 Upload Image"** tab
2. Click upload area or drag an image
3. Click **"🔍 Analyze Image"**
4. See results with annotated image and metrics

Option B: Monitor Live Webcam

1. Click **"📹 Webcam Monitor"** tab
2. Watch real-time EAR and MAR values
3. Alerts appear when drowsiness or yawning detected

What Do the Numbers Mean?

Metric	Normal Range	Drowsy Range
EAR (Eye Aspect Ratio)	> 0.25	< 0.25
MAR (Mouth Aspect Ratio)	< 0.70	> 0.70

- **EAR** = How open your eyes are (lower = closed eyes = drowsy)
- **MAR** = How open your mouth is (higher = wide open = yawning)

Common Issues & Solutions

❌ "Failed to analyze image"

- Check backend is running on `http://localhost:8000`
- Check browser console for error details
- Try a different image format (JPG, PNG)

❌ "Connecting to webcam feed..." (stuck)

- Backend webcam loop needs to start
- Restart backend: `uvicorn main:app --reload --port 8000`
- Check system permissions for webcam access

❌ Cannot access `http://localhost:5173`

- Frontend may not have started
- Try: `npm run dev` in frontend folder
- Check if port 5173 is in use

❌ Module errors after npm install

```
# Clear and reinstall
rm -r node_modules
npm install
npm run dev
```

Project Structure

```
Driver_Drowsiness_Detection_System/
├── backend/
│   ├── drowsiness.py      # ML model + API
│   ├── main.py            # FastAPI app
│   └── requirements.txt    # Python dependencies
├── frontend/
│   ├── src/
│   │   ├── components/    # React components
│   │   ├── App.jsx         # Main app
│   │   └── index.css       # Styles
│   ├── package.json        # npm dependencies
│   └── vite.config.js      # Vite config
```

Key Features

📊 Real-Time Metrics

- Eye Aspect Ratio (EAR)
- Mouth Aspect Ratio (MAR)

🚨 Smart Alerts

- Drowsiness detection
- Yawning detection
- Combined risk assessment

📸 Analysis Methods

- Single image upload
- Live webcam monitoring
- Annotated result images

Customization

Change Backend URL

Edit in `src/App.jsx` and `src/components/ImageUpload.jsx`:

```
const response = await fetch('http://your-api-url/endpoint')
```

Change Polling Frequency

In `src/App.jsx` (line ~20), change interval:

```
}, 500)  // Currently 500ms - adjust as needed
```

Change Theme Colors

Edit CSS variables in `src/App.css`:

```
--primary-color: #667eea;
--danger-color: #f56565;
--success-color: #48bb78;
```

Development Commands

```
# Start dev server with hot reload
npm run dev

# Build for production
npm run build

# Preview production build
npm run preview

# Run linting
npm run lint
```

Testing

Test Image Upload

1. Save a photo of yourself
2. Click Upload Image
3. Analyze it
4. Check the results

Test Webcam

1. Click Webcam Monitor
2. Wait for status to load
3. Try blinking, closing eyes, yawning
4. Watch metrics change in real-time

Next Steps

- Explore different images with varying drowsiness levels
- Test yawning detection
- Monitor metrics patterns
- Deploy to cloud (see CONFIGURATION.md)

Need More Help?

- Check `FRONTEND_README.md` for detailed documentation
- Check `CONFIGURATION.md` for advanced setup
- Look at backend `drowsiness.py` for ML details
- Check browser DevTools Console for error messages

Happy detecting! 🚗👁️