Guard Face Recognition Attendance System - API Documentation

Base URL

Copy

```
https://[your-ngrok-url].ngrok-free.app
```

1. Mark Attendance Endpoint

Endpoint: `POST /verify` **Purpose:** Capture guard's face image and verify attendance

Request:

- Method: `POST`
- Content-Type: `multipart/form-data`
- Body Parameter: `file` (Image file)

dart
Copy

```dart
// Example using Dio
final formData = FormData.fromMap({
  'file': await MultipartFile.fromFile(
    imageFile.path,
    filename: 'image.jpg',
    contentType: MediaType('image', 'jpeg'),
  ),
});

final response = await dio.post('/verify', data: formData);
```

Success Response (200):

json
Copy

```json
{
    "status": "success",
    "message": "Attendance marked successfully",
    "guard": {
        "id": "6727b282f9c1a72d308161a8",
        "name": "Jahan Zaib's Workspace",
        "role": "Civilian",
        "fatherName": "Dalton Sanders",
        "bloodGroup": "AB+",
        "mobileNumber": "123456",
        "picture":
"https://res.cloudinary.com/dolfkczwp/image/upload/..."
    },
    "confidence": 0.95,
    "anti_spoof_score": 0.98,
    "attendance_id": "507f1f77bcf86cd799439011"
}
```

Duplicate Attendance Response:

json
Copy

```
{
    "status": "info",
    "message": "Attendance already marked within last 30 minutes",
    "attendance_id": "507f1f77bcf86cd799439011",
    "guard": {
        "id": "6727b282f9c1a72d308161a8",
        "name": "Jahan Zaib's Workspace",
        "role": "Civilian"
    }
}
```

Error Responses:

json
Copy

```
{
    "status": "error",
    "message": "Spoof detected",
    "anti_spoof_score": 0.5
}
```

json
Copy

```
{
    "status": "error",
    "message": "No face detected"
}
```

json
Copy

```json
{
    "status": "error",
    "message": "Face not recognized"
}
```

Complete Flutter Implementation

1. Required Dependencies

Add these to your `pubspec.yaml`:

yaml
Copy

```yaml
dependencies:
  dio: ^5.0.0
  image_picker: ^0.8.6
  http_parser: ^4.0.2
```

2. Service Implementation

dart
Copy

```dart
class AttendanceService {
  final Dio dio;

  AttendanceService(String baseUrl)
      : dio = Dio(BaseOptions(
          baseUrl: baseUrl,
          connectTimeout: Duration(seconds: 30),
          receiveTimeout: Duration(seconds: 30),
        ));

  Future<Map<String, dynamic>> markAttendance(File imageFile) async {
    try {
      FormData formData = FormData.fromMap({
        'file': await MultipartFile.fromFile(
          imageFile.path,
          filename: 'image.jpg',
          contentType: MediaType('image', 'jpeg'),
        ),
      });

      final response = await dio.post('/verify', data: formData);
      return response.data;

    } catch (e) {
      throw Exception('Failed to mark attendance: $e');
    }
  }
}
```

3. Implementation Example

dart
Copy

```
class AttendanceScreen extends StatefulWidget {
  @override
  _AttendanceScreenState createState() => _AttendanceScreenState();
}

class _AttendanceScreenState extends State<AttendanceScreen> {
  final AttendanceService _service =
AttendanceService('YOUR_NGROK_URL');
  bool _isLoading = false;

  Future<void> _markAttendance() async {
    try {
      setState(() => _isLoading = true);

      // Capture image
      final ImagePicker picker = ImagePicker();
      final XFile? image = await picker.pickImage(
        source: ImageSource.camera,
        preferredCameraDevice: CameraDevice.front,
        imageQuality: 85,
      );

      if (image == null) return;

      // Mark attendance
      final result = await _service.markAttendance(File(image.path));

      if (result['status'] == 'success') {
        // Handle success
        showSuccessDialog(result['guard']['name']);
      } else {
        // Handle error
        showErrorDialog(result['message']);
      }

    } catch (e) {
      showErrorDialog(e.toString());
    } finally {
      setState(() => _isLoading = false);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Mark Attendance')),
      body: Center(
        child: _isLoading
          ? CircularProgressIndicator()
          : ElevatedButton(
              onPressed: _markAttendance,
              child: Text('Mark Attendance'),
            ),
      ),
    );
  }
}
```

Important Notes

1. Camera Configuration:

- Use front camera for better face capture
- Set image quality to 85 for optimal size/quality balance
- Ensure good lighting conditions

2. Error Handling:

dart
Copy

```
// Common error messages to handle
switch (result['message']) {
  case 'Spoof detected':
    // Show message about potential spoofing attempt
    break;
  case 'No face detected':
    // Guide user to position face properly
    break;
  case 'Face not recognized':
    // Inform user they are not registered
    break;
}
```

3. Required Permissions:

xml
Copy

```xml
<!-- Android: AndroidManifest.xml -->
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />

<!-- iOS: Info.plist -->
<key>NSCameraUsageDescription</key>
<string>Camera access is needed to capture face for attendance</string>
```

4. Best Practices:

- Handle timeouts (30 seconds configured)
- Show loading states during API calls
- Provide clear feedback messages
- Handle offline scenarios
- Clear image file after use

Testing Steps

1. Set up dependencies
2. Configure permissions
3. Replace 'YOUR_NGROK_URL' with actual URL
4. Test all error scenarios
5. Verify camera configuration
6. Test duplicate attendance handling