This document contains: 1️⃣ **OpenAPI / Swagger-style API specification (MVP)** 2️⃣ **React component tree mapped screen-by-screen from wireframes**

Everything aligns strictly with the approved MVP scope.

---

# 1️⃣ OPENAPI / SWAGGER SPEC (MVP)

## 🔐 Authentication

### POST /auth/login

**Description:** Login user and issue JWT

**Request**

```
{
  "email": "hr@company.com",
  "password": "******"
}
```

**Response**

```
{
  "token": "jwt_token",
  "user": {
    "id": "u_123",
    "role": "HR",
    "companyId": "c_001"
  }
}
```

---

## 👤 Candidate Management

### POST /candidates

**Create new candidate verification**

**Request**

```
{
  "name": "Ravi Kumar",
  "email": "ravi@email.com",
  "role": "Software Engineer"
}
```

**Response**

```
{
  "candidateId": "cand_001",
  "status": "CREATED"
}
```

### GET /candidates

**List candidates**

---

## 📄 Document Verification

### POST /documents/upload

**Upload document (S3 signed URL)**

### POST /verify/document

**Run OCR + fraud detection**

**Response**

```
{
  "documentId": "doc_001",
  "authenticityScore": 76,
  "riskLevel": "MEDIUM",
  "signals": ["FONT_MISMATCH"]
}
```

### GET /verify/document/{documentId}

---

## 🤳 Identity Verification

### POST /verify/identity

**Response**

```
{
  "faceMatchScore": 88,
  "status": "MATCH",
  "quality": "GOOD"
}
```

---

## 🏠 Address Intelligence

### POST /verify/address

#### Request

```json
{
  "address": "Madhapur, Hyderabad, 500081"
}
```

#### Response

```json
{
  "riskScore": 6,
  "riskLevel": "MEDIUM",
  "signals": ["HIGH_DENSITY_RENTAL"]
}
```

---

## 🧮 Trust Score

### POST /trust-score/calculate

#### Response

```json
{
  "trustScore": 72,
  "riskBucket": "MEDIUM",
  "breakdown": {
    "document": 30,
    "face": 25,
    "address": 10,
    "employment": 7
  }
}
```

### GET /trust-score/{candidateId}

---

## 📄 Reports

### GET /reports/{candidateId}

---

### 🔑 API & Usage

GET /api-keys

GET /api-usage

---

## 2️⃣ REACT COMPONENT TREE (MVP)

### 🧱 App Root

```
<App>
 ├── <AuthProvider />
 ├── <Router />
 └── <AppShell />
```

---

### 🔐 Login Page

```
<LoginPage>
 ├── <AuthLayout>
 ├── <TextInput email />
 ├── <PasswordInput />
 └── <PrimaryButton />
```

---

### 📊 Dashboard

```
<DashboardPage>
 ├── <Sidebar />
 ├── <KPICards />
 ├── <VerificationTable />
 └── <NewVerificationButton />
```

---

### 👤 Candidate Flow

```
<CandidateCreatePage>
 ├── <CandidateForm />
 └── <Stepper />
```

---

## Document AI

```
<DocumentUploadPage>
 ├── <FileUploader />
 ├── <SubmitButton />
 │

<DocumentResultPage>
 ├── <ScoreBar />
 ├── <SignalList />
 └── <ContinueButton />
```

---

## Identity Verification

```
<IdentityPage>
 ├── <SelfieCapture />
 ├── <FaceScore />
 └── <QualityAlert />
```

---

## Address Intelligence

```
<AddressPage>
 ├── <AddressText />
 ├── <MapEmbed />
 ├── <RiskScore />
 └── <SignalList />
```

---

## Trust Score

```
<TrustScorePage>
 ├── <HeroScore />
 ├── <ScoreBreakdown />
 ├── <RecommendationBox />
 └── <DownloadReport />
```

---

## Reports

```
<ReportPage>
 ├── <PDFPreview />
 └── <DownloadButton />
```

---

## 🔑 API Screen

```
<ApiPage>
 ├── <ApiKeyCard />
 ├── <UsageTable />
```

---

## ☑ Why this is execution-ready

• Backend contracts locked • Frontend architecture clear • Easy parallel dev (FE + BE) • Investor-grade technical clarity

---

If you want next: - Swagger YAML export - Backend folder structure - Frontend state management plan - Investor demo narration