# Recon-0 API Contract - Complete Backend Blueprint

**Document Version:** 3.0 (Production-Ready)
**For:** Sujal (Backend Developer)
**From:** Asim (Frontend Developer)
**Date:** September 8, 2025

This document serves as the complete API contract and blueprint for the Recon-0 Java Spring Boot backend. The frontend and fully functional mock API (`recon0-mock-api/server.js`) have been completed. Your task is to build the real backend service that exactly matches the endpoints, request/response structures, and logic defined in this contract.

## Table of Contents

---

## 1. Architecture Overview

### Core Technologies

- **Backend:** Java Spring Boot
- **Database:** PostgreSQL (use `recon-0-lite-V2.sql` schema)
- **Authentication:** JWT-based system (implement from scratch)
- **File Storage:** Cloudinary (recommended) or similar cloud provider
- **AI Integration:** Local LM Studio server at `http://localhost:1234/v1/`

### Base URL

All endpoints are prefixed with: `/api/v1`

### Standard Response Format

```json
{
  "success": true|false,
  "data": {...} | [...], // Present on success
  "message": "...",      // Present on error or info
  "error": "..."         // Present on error (optional)
}
```

---

## 2. Authentication System

### JWT Implementation

- Generate JWT tokens with payload: `{ id, role, email }`
- All protected endpoints require `Authorization: Bearer <JWT>` header
- Token validation middleware required

### Role-Based Access Control

- **hacker**: Can submit reports, view programs
- **organization**: Can manage programs, triage reports
- **admin**: Platform administration access

---

## 3. Common Data Types

### User Profile Object

```json
```

```json
{
  "id": "uuid",
  "email": "string",
  "username": "string",
  "full_name": "string",
  "role": "hacker|organization|admin",
  "status": "Active|Suspended",
  "reputation_points": "integer",
  "avatar_url": "string|null",
  "bio": "string|null",
  "created_at": "ISO_DATE_STRING"
}
```

## Program Object

json

```json
{
  "id": "uuid",
  "organization_id": "uuid",
  "org_name": "string",
  "title": "string",
  "description": "string",
  "policy": "string",
  "scope": "string",
  "out_of_scope": "string",
  "min_bounty": "integer",
  "max_bounty": "integer",
  "tags": ["string"],
  "org_logo_url": "string|null"
}
```

## Report Object

json

```json
{
  "id": "uuid",
  "program_id": "uuid",
  "program_name": "string",
  "reporter_id": "uuid",
  "title": "string",
  "severity": "Critical|High|Medium|Low",
  "status": "New|Triaging|Accepted|Resolved|Duplicate|Invalid",
  "description": "string",
  "steps_to_reproduce": "string",
  "impact": "string",
  "created_at": "ISO_DATE_STRING",
  "attachments": [AttachmentObject]
}
```

### Attachment Object

json

```json
{
  "id": "uuid",
  "file_url": "string",
  "file_name": "string",
  "file_type": "string",
  "uploaded_at": "ISO_DATE_STRING"
}
```

---

## 4. Authentication Endpoints

### Register User

**Endpoint:** `POST /auth/register`

**Access:** Public

**Description:** Creates a new user account with auto-login

**Request Body:**

json

```
{
  "email": "user@example.com",
  "password": "strongpassword123",
  "username": "unique_username",
  "fullName": "User Full Name",
  "role": "hacker|organization|admin"
}
```

**Response (201):**

```json
{
  "success": true,
  "message": "User registered successfully!",
  "token": "jwt_token_string",
  "user": {
    "id": "uuid",
    "username": "unique_username",
    "role": "hacker"
  }
}
```

**Error Cases:**

- 400: Email already exists, Username taken
- 422: Validation errors

### Login User

**Endpoint:** `POST /auth/login`
**Access:** Public
**Description:** Authenticates user and returns JWT token

**Request Body:**

```json
{
  "email": "user@example.com",
  "password": "password123"
}
```

**Response (200):**

```json
```

```json
{
  "success": true,
  "message": "Login successful!",
  "token": "jwt_token_string",
  "user": {
    "id": "uuid",
    "username": "username",
    "role": "hacker"
  }
}
```

**Error Cases:**

- 401: Invalid credentials
- 403: Account suspended

---

## 5. Profile Management

### Get Current User Profile

**Endpoint:** `GET /profile`
**Access:** Protected
**Description:** Returns complete profile of authenticated user

**Response (200):**

```json
{
  "success": true,
  "data": {
    "id": "uuid",
    "email": "user@example.com",
    "username": "username",
    "full_name": "User Name",
    "role": "hacker",
    "status": "Active",
    "reputation_points": 150,
    "avatar_url": "https://cloudinary.com/image.jpg",
    "bio": "Security researcher and bug hunter",
    "created_at": "2025-09-01T10:00:00Z"
  }
}
```

### Update User Profile

**Endpoint:** `PUT /profile`

**Access:** Protected

**Description:** Updates mutable profile fields

**Request Body:**

```json
{
  "fullName": "Updated Name",
  "username": "new_username",
  "bio": "Updated bio text"
}
```

**Response (200):**

```json
{
  "success": true,
  "data": {
    // Complete updated profile object
  }
}
```

**Error Cases:**

- 400: Username already taken
- 422: Validation errors

### Get User Stats

**Endpoint:** `GET /stats`

**Access:** Protected

**Description:** Returns statistics for authenticated user

**Response (200):**

```json
```

```
{
  "success": true,
  "data": {
    "reputation_points": 150,
    "reports_submitted": 12,
    "reports_accepted": 8,
    "bounties_earned": 2500
  }
}
```

---

## 6. File Upload System

### Upload Avatar

**Endpoint:** `POST /upload/avatar`
**Access:** Protected
**Content-Type:** `multipart/form-data`
**Description:** Uploads user avatar image

#### Form Data:

- `file`: Image file (JPEG, PNG, WebP)

#### Response (200):

```json
{
  "success": true,
  "message": "Avatar uploaded successfully!",
  "secure_url": "https://cloudinary.com/avatar.jpg"
}
```

### Upload Organization Logo

**Endpoint:** `POST /organization/upload/logo`
**Access:** Protected (organization role)
**Content-Type:** `multipart/form-data`
**Description:** Uploads organization logo

#### Form Data:

- `file`: Image file

#### Response (200):

```json
{
  "success": true,
  "message": "Logo uploaded successfully!",
  "secure_url": "https://cloudinary.com/logo.jpg"
}
```

### Upload Attachment

**Endpoint:** `POST /upload/attachment`
**Access:** Protected
**Content-Type:** `multipart/form-data`
**Description:** Uploads file attachment (for reports/messages)

#### Form Data:

- `file`: Any file type (images, documents, videos)

#### Response (200):

```json
{
  "success": true,
  "message": "File uploaded successfully!",
  "url": "https://cloudinary.com/file.pdf",
  "name": "original_filename.pdf",
  "type": "application/pdf"
}
```

---

## 7. Program Management

### Get All Programs

**Endpoint:** `GET /programs`
**Access:** Protected
**Description:** Returns all active bug bounty programs

#### Response (200):

```json
```

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "organization_id": "uuid",
      "org_name": "CyberCorp",
      "title": "Web Application Security Test",
      "min_bounty": 500,
      "max_bounty": 5000,
      "tags": ["web", "api", "critical"],
      "org_logo_url": "https://cloudinary.com/logo.jpg"
    }
  ]
}
```

### Get Program Details

**Endpoint:** `GET /programs/:id`

**Access:** Protected

**Description:** Returns complete details of a specific program

**Response (200):**

```json
{
  "success": true,
  "data": {
    "id": "uuid",
    "organization_id": "uuid",
    "org_name": "CyberCorp",
    "title": "Web Application Security Test",
    "description": "Comprehensive security assessment...",
    "policy": "Please follow responsible disclosure...",
    "scope": "All subdomains of *.example.com",
    "out_of_scope": "staging.example.com excluded",
    "min_bounty": 500,
    "max_bounty": 5000,
    "tags": ["web", "api"],
    "org_logo_url": "https://cloudinary.com/logo.jpg"
  }
}
```

---

## 8. Report Management

### Submit Report

**Endpoint:** `POST /reports`

**Access:** Protected (hacker role)

**Description:** Submits new vulnerability report

#### Request Body:

```json
{
  "programId": "uuid",
  "title": "XSS Vulnerability in Profile Page",
  "severity": "Medium",
  "description": "A stored XSS vulnerability exists...",
  "steps_to_reproduce": "1. Navigate to profile\n2. Enter script tag...",
  "impact": "Attackers can steal user sessions...",
  "attachments": [
    {
      "url": "https://cloudinary.com/screenshot.png",
      "name": "proof_of_concept.png",
      "type": "image/png"
    }
  ]
}
```

#### Response (201):

```json
```

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "program_id": "uuid",
    "program_name": "Web App Test",
    "reporter_id": "uuid",
    "title": "XSS Vulnerability in Profile Page",
    "severity": "Medium",
    "status": "New",
    "description": "A stored XSS vulnerability exists...",
    "steps_to_reproduce": "1. Navigate to profile...",
    "impact": "Attackers can steal user sessions...",
    "created_at": "2025-09-08T15:30:00Z",
    "attachments": [...]
  }
}
```

### Get User Reports

**Endpoint:** `GET /my-reports`

**Access:** Protected (hacker role)

**Description:** Returns all reports submitted by authenticated user

**Response (200):**

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "program_name": "Web App Test",
      "title": "XSS Vulnerability",
      "severity": "Medium",
      "status": "Accepted",
      "created_at": "2025-09-08T15:30:00Z"
    }
  ]
}
```

### Get Report Details

**Endpoint:** `GET /reports/:id`

**Access:** Protected (report owner or program owner)

**Description:** Returns complete report details with attachments

**Response (200):**

```json
{
  "success": true,
  "data": {
    "id": "uuid",
    "program_id": "uuid",
    "program_name": "Web App Test",
    "title": "XSS Vulnerability in Profile Page",
    "severity": "Medium",
    "status": "New",
    "description": "Detailed vulnerability description...",
    "steps_to_reproduce": "Step by step reproduction...",
    "impact": "Impact assessment...",
    "created_at": "2025-09-08T15:30:00Z",
    "attachments": [
      {
        "id": "uuid",
        "file_url": "https://cloudinary.com/proof.png",
        "file_name": "proof_of_concept.png",
        "file_type": "image/png",
        "uploaded_at": "2025-09-08T15:30:00Z"
      }
    ]
  }
}
```

### Get Report Messages

**Endpoint:** `GET /reports/:reportId/messages`
**Access:** Protected (report stakeholders)
**Description:** Returns formal communication log for a report

**Response (200):**

```json
```

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "report_id": "uuid",
      "sender_id": "uuid",
      "content": "Thank you for your submission. We are investigating...",
      "created_at": "2025-09-08T16:00:00Z",
      "attachments": [
        {
          "id": "uuid",
          "file_url": "https://cloudinary.com/response.pdf",
          "file_name": "investigation_log.pdf",
          "file_type": "application/pdf"
        }
      ]
    }
  ]
}
```

---

## 9. Organization Features

### Get Organization Dashboard

**Endpoint:** `GET /organization/dashboard`

**Access:** Protected (organization role)

**Description:** Returns dashboard data for organization

#### Response (200):

```
json
```

```
{
  "success": true,
  "data": {
    "kpis": {
      "programCount": 3,
      "totalReports": 45,
      "newReports": 12
    },
    "recentReports": [
      {
        "id": "uuid",
        "title": "SQL Injection in Login",
        "severity": "High",
        "status": "New",
        "created_at": "2025-09-08T14:00:00Z"
      }
    ]
  }
}
```

### Get Organization Programs

**Endpoint:** `GET /organization/my-programs`

**Access:** Protected (organization role)

**Description:** Returns all programs created by organization

**Response (200):**

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "title": "Web Application Test",
      "min_bounty": 500,
      "max_bounty": 5000,
      "tags": ["web", "api"],
      "org_logo_url": "https://cloudinary.com/logo.jpg"
    }
  ]
}
```

### Create Program

**Endpoint:** `POST /organization/my-programs`

**Access:** Protected (organization role)

**Description:** Creates new bug bounty program

**Request Body:**

```json
{
  "title": "API Security Assessment",
  "description": "Comprehensive API security testing program",
  "policy": "Follow responsible disclosure guidelines",
  "scope": "All API endpoints at api.example.com",
  "out_of_scope": "Internal development APIs excluded",
  "min_bounty": 250,
  "max_bounty": 3000,
  "tags": ["api", "security", "authentication"]
}
```

**Response (201):**

```json
{
  "success": true,
  "data": {
    "id": "uuid",
    "organization_id": "uuid",
    "org_name": "Organization Name",
    "title": "API Security Assessment",
    // ... complete program object
  }
}
```

### Get Organization Reports

**Endpoint:** `GET /organization/reports`

**Access:** Protected (organization role)

**Description:** Returns all reports submitted to organization's programs

**Response (200):**

```json
```

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "program_name": "Web App Test",
      "title": "XSS Vulnerability",
      "severity": "Medium",
      "status": "New",
      "created_at": "2025-09-08T15:30:00Z"
    }
  ]
}
```

### Update Report Status

**Endpoint:** `PATCH /organization/reports/:reportId`

**Access:** Protected (organization role, program owner)

**Description:** Updates report status (triage action)

**Request Body:**

```json
{
  "status": "Accepted"
}
```

**Response (200):**

```json
{
  "success": true,
  "data": {
    // Complete updated report object
  }
}
```

### Send Report Message

**Endpoint:** `POST /reports/:reportId/messages`

**Access:** Protected (organization role, program owner)

**Description:** Sends formal reply to report

**Request Body:**

```json
{
  "content": "Thank you for your submission. We have confirmed the vulnerability and are working on a fix.",
  "attachments": [
    {
      "url": "https://cloudinary.com/patch_notes.pdf",
      "name": "fix_timeline.pdf",
      "type": "application/pdf"
    }
  ]
}
```

**Response (201):**

```json
{
  "success": true,
  "data": {
    "id": "uuid",
    "report_id": "uuid",
    "sender_id": "uuid",
    "content": "Thank you for your submission...",
    "created_at": "2025-09-08T16:30:00Z",
    "attachments": [...]
  }
}
```

### Get Program Analytics

**Endpoint:** `GET /organization/programs/:programId/analytics`
**Access:** Protected (organization role, program owner)
**Description:** Returns analytics data for specific program

**Response (200):**

```json
```

```
{
  "success": true,
  "data": {
    "programTitle": "Web Application Test",
    "kpis": {
      "totalReports": 132,
      "resolvedReports": 98,
      "avgTimeToBountyDays": 14,
      "totalPaidOut": 25400
    },
    "reportsBySeverity": [
      { "severity": "Critical", "count": 12 },
      { "severity": "High", "count": 35 },
      { "severity": "Medium", "count": 68 },
      { "severity": "Low", "count": 17 }
    ],
    "submissionTrend": [
      { "date": "Aug 1", "count": 15 },
      { "date": "Aug 8", "count": 22 },
      { "date": "Aug 15", "count": 18 }
    ]
  }
}
```

---

## 10. Admin Features

### Get All Users

**Endpoint:** `GET /admin/users`
**Access:** Protected (admin role)
**Description:** Returns list of all platform users

#### Response (200):

```json
```

```
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "username": "hacker123",
      "full_name": "John Doe",
      "email": "john@example.com",
      "role": "hacker",
      "status": "Active",
      "reputation_points": 1250,
      "created_at": "2025-08-15T10:00:00Z"
    }
  ]
}
```

### Update User Status

**Endpoint:** `PATCH /admin/users/:userId/status`

**Access:** Protected (admin role)

**Description:** Updates user status (suspend/reactivate)

#### Request Body:

```json
{
  "status": "Suspended"
}
```

#### Response (200):

```json
{
  "success": true,
  "data": {
    // Complete updated user object (without password)
  }
}
```

#### Business Rules:

- Admin cannot change their own status
- Status values: "Active", "Suspended"

### Get Platform Analytics

**Endpoint:** `GET /admin/analytics`

**Access:** Protected (admin role)

**Description:** Returns platform-wide analytics

**Response (200):**

```json
{
  "success": true,
  "data": {
    "kpis": {
      "totalUsers": 1250,
      "totalHackers": 980,
      "totalOrgs": 45,
      "totalReports": 3420,
      "totalPrograms": 128
    },
    "reportsByStatus": [
      { "status": "New", "count": 156 },
      { "status": "Accepted", "count": 890 },
      { "status": "Triaging", "count": 234 }
    ]
  }
}
```

---

## 11. Community Features

### Get Leaderboard

**Endpoint:** `GET /leaderboard`

**Access:** Protected

**Description:** Returns top hackers by reputation

**Response (200):**

```json
```

```
{
  "success": true,
  "data": [
    {
      "rank": 1,
      "username": "cyb3r_ninja",
      "reputation_points": 9850
    },
    {
      "rank": 2,
      "username": "glitch_hunter",
      "reputation_points": 9500
    }
  ]
}
```

### Get Notifications

**Endpoint:** `GET /notifications`

**Access:** Protected

**Description:** Returns user notifications

**Response (200):**

```json
{
  "success": true,
  "data": [
    {
      "id": 1,
      "user_id": "uuid",
      "type": "report_update",
      "message": "Your report 'XSS in Profile Page' was accepted.",
      "is_read": false,
      "created_at": "2025-09-08T15:30:00Z"
    }
  ]
}
```

### Get All Achievements

**Endpoint:** `GET /achievements`

**Access:** Protected

**Description:** Returns list of all possible achievements

**Response (200):**

```json
{
  "success": true,
  "data": [
    {
      "id": "ach-1",
      "name": "First Find",
      "description": "Submit your first valid report.",
      "icon": "fa-flag"
    }
  ]
}
```

### Get User Achievements

**Endpoint:** `GET /achievements/my`

**Access:** Protected

**Description:** Returns achievements earned by user

**Response (200):**

```json
{
  "success": true,
  "data": ["ach-1", "ach-3", "ach-5"]
}
```

---

## 12. AI Integration

### Enhance Report with AI

**Endpoint:** `POST /ai/enhance-report`

**Access:** Protected

**Description:** Uses local LLM to enhance report quality

**Request Body:**

```json
```

```
{
  "description": "Basic vulnerability description",
  "steps_to_reproduce": "Simple reproduction steps",
  "impact": "Basic impact statement"
}
```

**Response (200):**

```json
{
  "success": true,
  "data": {
    "description": "Enhanced professional vulnerability description...",
    "steps_to_reproduce": "1. Navigate to the target page...\n2. Enter malicious payload...",
    "impact": "This vulnerability poses significant security risks..."
  }
}
```

### AI Chatbot

**Endpoint:** `POST /ai/chat`

**Access:** Protected

**Description:** General Q&A chatbot for platform help

**Request Body:**

```json
{
  "question": "What is a bug bounty program?"
}
```

**Response (200):**

```json
{
  "success": true,
  "data": {
    "answer": "A bug bounty program is a crowdsourced cybersecurity initiative where organizations invite security resea
  }
}
```

### AI Integration Notes

- Connect to local LM Studio server at `http://localhost:1234/v1/`
- Handle server unavailability gracefully
- Use appropriate system prompts for different AI functions
- Implement retry logic and fallback responses

---

## 13. Error Handling

### Standard Error Response

json

```json
{
  "success": false,
  "message": "Human-readable error message",
  "error": "TECHNICAL_ERROR_CODE"
}
```

### HTTP Status Codes

- **200**: Success
- **201**: Created
- **400**: Bad Request (validation errors)
- **401**: Unauthorized (no/invalid token)
- **403**: Forbidden (insufficient permissions)
- **404**: Not Found
- **422**: Unprocessable Entity (business logic errors)
- **500**: Internal Server Error

### Common Error Scenarios

1. **Authentication Errors**: Invalid/expired JWT tokens
2. **Authorization Errors**: Insufficient role permissions
3. **Validation Errors**: Invalid input data format
4. **Business Logic Errors**: Violating application rules
5. **File Upload Errors**: Invalid file types, size limits
6. **AI Service Errors**: LM Studio server unavailable

### Implementation Requirements

1. **Database Schema**: Use the exact schema from `recon-0-lite-V2.sql`

2. **JWT Security**: Implement proper token signing and validation
3. **File Storage**: Integrate with Cloudinary for all file uploads
4. **Role Validation**: Strict role-based access control
5. **Data Validation**: Comprehensive input validation
6. **Error Logging**: Log all errors for debugging
7. **CORS Configuration**: Enable CORS for frontend communication

---

## Final Notes

This API contract represents the complete feature set of the Recon-0 platform as implemented in the working frontend and mock API. Every endpoint listed here is actively used by the frontend application.

The mock server (`server.js`) serves as your reference implementation - it contains the exact logic, data transformations, and business rules that should be replicated in your Java Spring Boot backend.

**Key Success Criteria:**

- All endpoints must match exactly (paths, methods, request/response formats)
- JWT authentication must work identically to the mock implementation
- File uploads must integrate with cloud storage and return public URLs
- Role-based access control must be strictly enforced
- AI features must connect to local LM Studio server

The frontend is complete and functional - your backend implementation should be a drop-in replacement for the mock API server.