N8N Recruitment & Hiring System

Complete Workflow Documentation

WORKFLOW 1: Job Posting Creation

Trigger: Manual or Scheduled
Purpose: Create and publish job posting on LinkedIn

Steps:

1. **Manual Trigger Node**
   - Start workflow manually when HR is ready to post
2. **Set Node - Prepare Job Details**
   - Input Fields:
     - Job Title
     - Job Description
     - Required Skills
     - Experience Required
     - Location
     - Salary Range
     - Application Link (company website)
   - Output: Structured job data
3. **LinkedIn Node - Create Post**
   - Action: Create Share/Post
   - Content: Format job details with hashtags
   - Include: Application link to company website
   - Output: Post ID, Post URL
4. **Google Sheets Node - Log Job Posting**
   - Sheet: "Job_Postings"
   - Action: Append Row
   - Data: Job_ID, Title, LinkedIn_URL, Date_Posted, Status
   - Output: Confirmation
5. **Slack/Email Node - Notify HR Team**
   - Message: "Job posted successfully on LinkedIn"

- Include: LinkedIn post URL
- End Workflow

Output: Job posting live on LinkedIn with tracking ID

WORKFLOW 2: Application Reception & Storage

Trigger: Webhook (from company website)
Purpose: Receive and store candidate applications

Steps:

1. **Webhook Trigger Node**
   - Method: POST
   - Webhook URL: https://your-n8n.com/webhook/job-application
   - Expected Data:
     - candidate_name
     - email
     - phone
     - cv_file_url
     - job_id
     - application_date
2. **Set Node - Clean & Structure Data**
   - Normalize data format
   - Generate unique Candidate_ID (UUID)
   - Add timestamp
   - Set initial status: "Applied"
3. **HTTP Request Node - Download CV**
   - URL: {{$json["cv_file_url"]}}
   - Method: GET
   - Output: Binary CV file
4. **Google Drive Node - Upload CV**
   - Action: Upload File
   - Folder: "Candidate_CVs/[Job_ID]"
   - Filename: "[Candidate_ID]_[Name].pdf"

- Output: Google Drive file ID and shareable link

5. **Google Sheets Node - Store Application**
   - Sheet: "Candidates"
   - Action: Append Row
   - Data: All candidate info + CV_Link + Status
   - Output: Row number
6. **Gmail Node - Send Acknowledgment**
   - To: {{$json["email"]}}
   - Subject: "Application Received - [Job_Title]"
   - Body: "Thank you for applying. We'll review your application."
7. **HTTP Request Node - Trigger Screening Workflow**
   - URL: Webhook URL of Workflow 3
   - Method: POST
   - Body: Candidate data
   - End Workflow

Output: Application stored, CV uploaded, screening triggered

WORKFLOW 3: AI-Based CV Screening

Trigger: Webhook (from Workflow 2)
Purpose: Analyze CV against job requirements using AI

Steps:

1. **Webhook Trigger Node**
   - Receives: Candidate_ID, CV_Link, Job_ID, Email
2. **Google Sheets Node - Get Job Description**
   - Sheet: "Job_Postings"
   - Action: Lookup by Job_ID
   - Output: Full job description and requirements
3. **HTTP Request Node - Download CV Text**
   - URL: {{$json["cv_link"]}}
   - Parse PDF to text (or use Google Drive API)
4. **OpenAI Node - CV Analysis**

   - Model: GPT-4
   - Prompt Template:

5. Analyze this candidate's CV against the job requirements.
6.
7. JOB DESCRIPTION:
8. {{$json["job_description"]}}
9.
10. REQUIRED SKILLS:
11. {{$json["required_skills"]}}
12.
13. CANDIDATE CV:
14. {{$json["cv_text"]}}
15.
16. Provide JSON response:
17. {
18. "match_score": <0-100>,
19. "skills_match": ["matched_skill1", "matched_skill2"],
20. "missing_skills": ["skill1", "skill2"],
21. "experience_years": <number>,
22. "education_match": <boolean>,
23. "recommendation": "SHORTLIST" or "REJECT",
24. "reasoning": "brief explanation"
25. }
    - Output: AI analysis JSON
26. **Function Node - Calculate Final Score**
    - Parse AI response
    - Apply weighted scoring logic
    - Set threshold: 70/100 for shortlisting
27. **IF Node - Check Score**
    - Condition: {{$json["match_score"]}} >= 70
    - TRUE branch → Step 7

   - FALSE branch → Step 10

28. **Google Sheets Node - Update Status (Shortlisted)**
    - Sheet: "Candidates"
    - Action: Update Row (match Candidate_ID)
    - Fields: Status = "Shortlisted", Screening_Score, AI_Feedback
29. **HTTP Request Node - Trigger Exam Workflow**
    - URL: Webhook URL of Workflow 4
    - Method: POST
    - Body: Candidate data
    - End TRUE branch
30. **Google Sheets Node - Update Status (Rejected)**
    - Sheet: "Candidates"
    - Status = "Rejected - Screening", Screening_Score, Reason
31. **Gmail Node - Send Rejection Email**
    - To: Candidate email
    - Subject: "Application Update"
    - Body: Polite rejection message
    - End FALSE branch

Output: Candidates scored, shortlisted ones sent to exam stage

---

WORKFLOW 4: Online Exam Invitation

Trigger: Webhook (from Workflow 3)
Purpose: Generate and send exam links to shortlisted candidates

Steps:

1. **Webhook Trigger Node**
    - Receives: Candidate_ID, Name, Email, Job_ID
2. **Google Sheets Node - Get Exam Details**
    - Sheet: "Job_Postings"
    - Lookup: Exam template for this Job_ID
    - Output: Exam_ID, Duration, Passing_Score
3. **HTTP Request Node - Generate Exam Link**

   - Integration: HackerRank/Mettl/TestGorilla API
   - Endpoint: Create test invitation
   - Parameters:
     - candidate_email
     - test_id
     - validity: 48 hours
   - Output: Unique exam link

4. **Function Node - Set Expiry Time**
   - Calculate: Current time + 48 hours
   - Format: Human-readable datetime
5. **Google Sheets Node - Update Candidate Record**
   - Sheet: "Candidates"
   - Update: Status = "Exam Sent", Exam_Link, Exam_Expiry
6. **Gmail Node - Send Exam Invitation**
   - To: {{$json["email"]}}
   - Subject: "Online Assessment - [Job_Title]"
   - Body Template:
7. Dear {{$json["name"]}},
8.
9. Congratulations! You've been shortlisted for the next round.
10.
11. 📝 ONLINE ASSESSMENT DETAILS:
12. • Link: {{$json["exam_link"]}}
13. • Duration: {{$json["duration"]}} minutes
14. • Valid Until: {{$json["exam_expiry"]}}
15. • Passing Score: {{$json["passing_score"]}}%
16.
17. Important Instructions:
18. - Use a stable internet connection
19. - Complete in one sitting
20. - Do not refresh the page

21.

22. Good luck!

23.

24. Best regards,

25. HR Team

26. **Slack Node - Notify HR**
    - Channel: #recruitment
    - Message: "Exam sent to [Name] for [Job_Title]"
    - End Workflow

Output: Exam invitation sent, candidate record updated

WORKFLOW 5: Exam Result Processing

Trigger: Webhook (from exam platform)
Purpose: Receive exam scores and schedule GD for qualified candidates

Steps:

1. **Webhook Trigger Node**
    - Integration: Exam platform webhook
    - Receives: Candidate_ID, Exam_Score, Total_Score, Time_Taken, Completion_Status
2. **Function Node - Calculate Percentage**
    - Formula: (Exam_Score / Total_Score) * 100
    - Round to 2 decimals
3. **Google Sheets Node - Update Exam Score**
    - Sheet: "Candidates"
    - Match: Candidate_ID
    - Update: Exam_Score, Exam_Percentage, Exam_Completion_Date
4. **Google Sheets Node - Get Passing Criteria**
    - Sheet: "Job_Postings"
    - Lookup passing score for this job
5. **IF Node - Check Pass/Fail**
    - Condition: {{$json["exam_percentage"]}} >= {{$json["passing_score"]}}
    - TRUE → Step 6

- FALSE → Step 11

6. **Google Sheets Node - Update Status (Exam Passed)**
   - Status = "Exam Passed - Pending GD"
7. **HTTP Request Node - Trigger GD Workflow**
   - URL: Webhook URL of Workflow 6
   - Method: POST
   - Body: Candidate_ID, Name, Email, Exam_Score
   - End TRUE branch
8. **Google Sheets Node - Update Status (Exam Failed)**
   - Status = "Rejected - Exam", Final_Score
9. **Gmail Node - Send Exam Result (Failed)**
   - Subject: "Application Update"
   - Body: Thank you message with encouragement
   - End FALSE branch

Output: Exam results recorded, qualified candidates moved to GD stage

---

WORKFLOW 6: Group Discussion Scheduling

Trigger: Webhook (from Workflow 5)
Purpose: Schedule and invite candidates to GD round

Steps:

1. **Webhook Trigger Node**
   - Receives: Candidate_ID, Name, Email, Exam_Score
2. **Google Sheets Node - Get GD Batch Info**
   - Sheet: "GD_Schedule"
   - Find next available GD slot
   - Max participants per slot: 8-10
3. **Function Node - Assign to Batch**
   - Logic: Add candidate to batch
   - If batch full, create new batch
   - Generate GD_Batch_ID
4. **Google Calendar Node - Create GD Event**

  - Action: Create Event
  - Title: "Group Discussion - [Job_Title] - Batch [X]"
  - Duration: 60 minutes
  - Attendees: All batch candidates + HR panel
  - Output: Google Meet link

5. **Google Sheets Node - Update Candidate Record**
   - Sheet: "Candidates"
   - Update: Status = "GD Scheduled", GD_Date, GD_Batch_ID, GD_Link
6. **Gmail Node - Send GD Invitation**
   - To: {{$json["email"]}}
   - Subject: "Group Discussion Invitation - [Job_Title]"
   - Body:
7. Dear {{$json["name"]}},
8.
9. Excellent work on the exam! You're invited to the Group Discussion round.
10.
11. 📅 GD DETAILS:
12. • Date & Time: {{$json["gd_datetime"]}}
13. • Duration: 60 minutes
14. • Meeting Link: {{$json["gd_link"]}}
15. • Topic: Will be shared at the start
16.
17. PREPARATION TIPS:
18. - Test your camera and microphone
19. - Join 5 minutes early
20. - Be ready to discuss current industry trends
21.
22. See you there!
23.
24. Best regards,
25. HR Team

26. **Wait Node**
    - Wait until: {{$json["gd_datetime"]}} + 90 minutes
    - (Allows time for GD completion)
27. **Slack Node - Reminder to HR**
    - Message: "Submit GD scores for Batch [X]"
    - Include: Link to scoring form
    - End Workflow

Output: GD scheduled, invitations sent, HR reminded to score

WORKFLOW 7: GD Result Processing

Trigger: Webhook (HR submits GD scores via form)
Purpose: Process GD results and schedule interviews

Steps:

1. **Webhook Trigger Node**
    - Form Data: Candidate_ID, GD_Score, Communication_Rating, Leadership_Rating, HR_Comments
2. **Function Node - Calculate GD Final Score**
    - Weighted average of ratings
    - Threshold for interview: 7/10
3. **Google Sheets Node - Update GD Scores**
    - Sheet: "Candidates"
    - Update: GD_Score, GD_Feedback, GD_Completion_Date
4. **IF Node - Check GD Pass/Fail**
    - Condition: {{$json["gd_score"]}} >= 7
    - TRUE → Step 5
    - FALSE → Step 9
5. **Google Sheets Node - Update Status (GD Passed)**
    - Status = "GD Passed - Pending Interview"
6. **HTTP Request Node - Trigger Interview Workflow**
    - URL: Webhook URL of Workflow 8
    - Method: POST
    - Body: Candidate data

- End TRUE branch

7. **Google Sheets Node - Update Status (GD Failed)**
    - Status = "Rejected - GD"
8. **Gmail Node - Send GD Result (Failed)**
    - Polite rejection email
    - End FALSE branch

Output: GD scores recorded, successful candidates moved to interview

---

WORKFLOW 8: Interview Scheduling

Trigger: Webhook (from Workflow 7)
Purpose: Schedule one-on-one interviews with hiring managers

Steps:

1. **Webhook Trigger Node**
    - Receives: Candidate_ID, Name, Email, GD_Score
2. **Google Sheets Node - Get Interviewer Availability**
    - Sheet: "Interviewer_Schedule"
    - Find available time slot for this Job_ID's panel
3. **Calendly/Google Calendar API - Book Interview Slot**
    - Duration: 45 minutes
    - Attendees: Candidate + Hiring Manager + HR
    - Output: Meeting link and calendar invite
4. **Google Sheets Node - Update Candidate Record**
    - Sheet: "Candidates"
    - Update: Status = "Interview Scheduled", Interview_Date, Interview_Link, Interviewer_Name
5. **Gmail Node - Send Interview Invitation**
    - To: {{$json["email"]}}
    - Subject: "Personal Interview - [Job_Title]"
    - Body:
6. Dear {{$json["name"]}},
7.
8. Great performance in the GD! We'd like to invite you for a personal interview.

9.

10. 🎯 INTERVIEW DETAILS:
11. • Date & Time: {{$json["interview_datetime"]}}
12. • Duration: 45 minutes
13. • Interview Link: {{$json["interview_link"]}}
14. • Interviewer: {{$json["interviewer_name"]}} ({{$json["interviewer_title"]}})
15.

16. WHAT TO EXPECT:
17. - Technical discussion about your experience
18. - Behavioral questions
19. - Company culture fit assessment
20. - Q&A about the role
21.

22. Please confirm your attendance by replying to this email.
23.

24. Best of luck!
25.

26. Regards,
27. HR Team
28. **Gmail Node - Send to Interviewer**
    - To: Interviewer email
    - Attach: Candidate CV and scores
    - Subject: "Interview Scheduled - [Candidate_Name]"
29. **Wait Node**
    - Wait until: Interview datetime + 60 minutes
30. **Slack Node - Reminder to Interviewer**
    - Message: "Please submit interview feedback for [Candidate_Name]"
    - Include: Feedback form link
    - End Workflow

Output: Interview scheduled, both parties notified

WORKFLOW 9: Interview Feedback & Final Selection

Trigger: Webhook (Interviewer submits feedback form)
Purpose: Process interview feedback and make hiring decision

Steps:

1. **Webhook Trigger Node**
   - Form Data: Candidate_ID, Technical_Score, Cultural_Fit, Salary_Expectation, Recommendation (Select/Reject/Hold), Interviewer_Comments
2. **Google Sheets Node - Update Interview Feedback**
   - Sheet: "Candidates"
   - Update: All interview fields, Interview_Completion_Date
3. **Function Node - Calculate Final Hiring Score**
   - Formula: Weighted average of:
     - AI Screening: 15%
     - Exam: 25%
     - GD: 30%
     - Interview: 30%
   - Output: Final_Score (0-100)
4. **Google Sheets Node - Update Final Score**
   - Update: Final_Score field
5. **IF Node - Check Recommendation**
   - Condition: {{$json["recommendation"]}} === "Select"
   - TRUE → Step 6
   - FALSE → Step 12
6. **Google Sheets Node - Update Status (Selected)**
   - Status = "Selected - Offer Pending"
7. **Slack Node - Notify HR Leadership**
   - Channel: #hiring-approvals
   - Message: "Candidate selected: [Name] for [Job_Title]"
   - Include: Final score and interviewer recommendation
   - Request: Approval for offer generation
8. **Wait Node**
   - Wait for: Manual approval (button click in n8n)

9. **HTTP Request Node - Trigger Offer Generation Workflow**
   - URL: Webhook URL of Workflow 10
   - Method: POST
   - Body: Candidate_ID, Salary_Expectation
   - End TRUE branch
10. **IF Node - Check Rejection Reason**
    - Condition: Recommendation = "Reject" or "Hold"
11. **Google Sheets Node - Update Status (Rejected/Hold)**
    - Status = "Rejected - Interview" or "On Hold"
12. **Gmail Node - Send Result Email**
    - If Reject: Polite rejection
    - If Hold: "We'll get back to you"
    - End FALSE branch

Output: Final decision made, offer process initiated or candidate rejected

---

WORKFLOW 10: Offer Letter Generation & Sending

Trigger: Webhook (from Workflow 9 after approval)
Purpose: Generate offer letter and send to selected candidate

Steps:

1. **Webhook Trigger Node**
   - Receives: Candidate_ID, Salary_Expectation
2. **Google Sheets Node - Get Full Candidate Details**
   - Sheet: "Candidates"
   - Fetch: Name, Email, Phone, Address, All_Scores
3. **Google Sheets Node - Get Job & Compensation Details**
   - Sheet: "Job_Postings"
   - Fetch: Job_Title, Department, Reporting_Manager, Base_Salary_Range
4. **Function Node - Calculate Offer Components**
   - Base Salary (based on expectation & budget)
   - Benefits breakdown
   - Start date (30 days from now)

    - Offer expiry (7 days from today)

5. **Google Docs API Node - Generate Offer Letter**
    - Template: "Offer_Letter_Template" (pre-created in Google Docs)
    - Replace placeholders:
        - {{candidate_name}}
        - {{job_title}}
        - {{base_salary}}
        - {{start_date}}
        - {{offer_expiry}}
        - etc.
    - Output: Document ID
6. **Google Drive API Node - Export as PDF**
    - Convert Google Doc to PDF
    - Filename: "Offer_Letter_[Candidate_Name]_[Date].pdf"
7. **Google Drive Node - Move to Offers Folder**
    - Folder: "Offer_Letters/[Year]"
    - Set sharing: Private
8. **Google Sheets Node - Update Status**
    - Sheet: "Candidates"
    - Update: Status = "Offer Sent", Offer_Date, Offer_Salary, Offer_Expiry
9. **Gmail Node - Send Offer Email**
    - To: {{$json["email"]}}
    - Subject: "Job Offer - [Job_Title] at [Company_Name]"
    - Attachment: Offer letter PDF
    - Body:
10. Dear {{$json["name"]}},
11.
12. Congratulations! 🎉
13.
14. We are delighted to offer you the position of {{$json["job_title"]}} at [Company Name].
15.

16. Please find attached your formal offer letter with complete details about:
17. • Compensation & Benefits
18. • Start Date: {{$json["start_date"]}}
19. • Reporting Structure
20. • Company Policies
21.
22. TO ACCEPT THIS OFFER:
23. 1. Review the attached document carefully
24. 2. Sign and return the document by {{$json["offer_expiry"]}}
25. 3. Submit required documents (list attached)
26.
27. If you have any questions, please don't hesitate to reach out.
28.
29. We're excited to have you join our team!
30.
31. Warm regards,
32. [HR Manager Name]
33. HR Department
34. [Company Name]
35. **Gmail Node - Send to HR & Manager**
    - CC: HR Head, Hiring Manager
    - Subject: "Offer Extended - [Candidate_Name]"
    - Body: Offer summary
36. **Slack Node - Celebrate!**
    - Channel: #general
    - Message: "🎉 Offer extended to [Name] for [Job_Title]! Welcome to the team!"
37. **Wait Node**
    - Wait until: Offer expiry date
38. **HTTP Request Node - Check Offer Status**
    - Query Google Sheets for offer acceptance status
39. **IF Node - Offer Accepted?**

  - If YES: Send onboarding workflow trigger
  - If NO: Send reminder or close
  - End Workflow

Output: Offer letter generated, sent, and tracked

AUXILIARY WORKFLOWS

WORKFLOW 11: Daily Status Report

Trigger: Schedule (Daily at 9 AM)
Purpose: Send recruitment pipeline report to HR team

Steps:

1. **Schedule Trigger** - Daily at 9:00 AM
2. **Google Sheets Node - Query All Active Candidates**
   - Count by status:
     - Applied
     - Screening
     - Shortlisted
     - Exam Sent
     - GD Scheduled
     - Interview Scheduled
     - Offer Sent
3. **Function Node - Format Report**
   - Create summary statistics
   - Identify bottlenecks
4. **Slack/Email Node - Send Report**
   - Recipients: HR team
   - Include: Charts/tables
   - End Workflow

WORKFLOW 12: Reminder for Pending Tasks

Trigger: Schedule (Every 6 hours)
Purpose: Remind HR to complete pending evaluations

Steps:

1. **Schedule Trigger** - Every 6 hours
2. **Google Sheets Node - Find Pending Tasks**
   - GD scores not submitted (>24 hours)
   - Interview feedback not submitted (>12 hours)
   - Offers not responded (3 days before expiry)
3. **Loop Node - For each pending task**
4. **Slack Node - Send Reminder**
   - Tag relevant person
   - Include candidate details
   - End Workflow

WORKFLOW 13: Candidate Communication Log

Trigger: Webhook (from any email-sending workflow)
Purpose: Log all communications with candidates

Steps:

1. **Webhook Trigger**
   - Receives: Candidate_ID, Email_Type, Timestamp, Content
2. **Google Sheets Node - Append to Communication Log**
   - Sheet: "Communication_History"
   - Data: All fields
3. **End Workflow**

INTER-WORKFLOW CONNECTIONS

WORKFLOW 1 (Job Posting)

↓

[Candidate applies on website]

↓

WORKFLOW 2 (Application Reception) → Triggers → WORKFLOW 3

↓

WORKFLOW 3 (AI Screening) → Triggers → WORKFLOW 4

↓

WORKFLOW 4 (Exam Invitation) → Waits for exam completion

↓

[Exam platform sends results]

↓

WORKFLOW 5 (Exam Processing) → Triggers → WORKFLOW 6

↓

WORKFLOW 6 (GD Scheduling) → Waits for HR input

↓

[HR submits GD scores]

↓

WORKFLOW 7 (GD Processing) → Triggers → WORKFLOW 8

↓

WORKFLOW 8 (Interview Scheduling) → Waits for interviewer input

↓

[Interviewer submits feedback]

↓

WORKFLOW 9 (Final Selection) → Triggers → WORKFLOW 10

↓

WORKFLOW 10 (Offer Generation)

↓

[End: Candidate hired or recruitment continues]

DATABASE STRUCTURE

Google Sheets Tables:

Sheet 1: Job_Postings

- Job_ID (Primary Key)
- Title
- Description
- Required_Skills
- LinkedIn_Post_URL
- Date_Posted

- Status (Active/Closed)
- Exam_Template_ID
- Passing_Score

Sheet 2: Candidates

- Candidate_ID (Primary Key)
- Job_ID (Foreign Key)
- Name
- Email
- Phone
- CV_Link
- Application_Date
- Status (Current stage)
- Screening_Score
- Exam_Score
- GD_Score
- Interview_Score
- Final_Score
- Offer_Salary
- Last_Updated

Sheet 3: Communication_History

- Log_ID
- Candidate_ID
- Email_Type
- Sent_Date
- Status (Delivered/Opened/Clicked)

Sheet 4: GD_Schedule

- Batch_ID
- Date_Time
- Candidates (Array)
- HR_Panel
- Status

Sheet 5: Interviewer_Schedule

- Interviewer_ID
- Name
- Available_Slots
- Job_Assignments

WEBHOOK URLs TO CONFIGURE

1. **Application Form**: Workflow 2 webhook
2. **Exam Platform**: Workflow 5 webhook
3. **GD Score Form**: Workflow 7 webhook
4. **Interview Feedback Form**: Workflow 9 webhook

EXTERNAL INTEGRATIONS NEEDED

1. **LinkedIn API** - For posting jobs
2. **Google Workspace** (Drive, Docs, Sheets, Calendar, Gmail)
3. **OpenAI API** - For CV screening
4. **Exam Platform API** (HackerRank/Mettl/TestGorilla)
5. **Slack API** - For team notifications
6. **Calendly API** (optional) - For interview scheduling

TESTING CHECKLIST

- [ ] Test Workflow 1: Job post created on LinkedIn
- [ ] Test Workflow 2: Application received and stored
- [ ] Test Workflow 3: AI screening works correctly
- [ ] Test Workflow 4: Exam invitation email sent
- [ ] Test Workflow 5: Exam results processed
- [ ] Test Workflow 6: GD scheduled correctly
- [ ] Test Workflow 7: GD scores recorded
- [ ] Test Workflow 8: Interview scheduled
- [ ] Test Workflow 9: Final selection logic
- [ ] Test Workflow 10: Offer letter generated

- [ ] Test inter-workflow triggers
- [ ] Test error handling
- [ ] Test email deliverability

DEPLOYMENT STEPS

1. Create all 13 workflows in n8n
2. Set up Google Sheets database structure
3. Create email templates in Gmail
4. Configure all webhook URLs
5. Set up API credentials for all integrations
6. Test each workflow independently
7. Test full end-to-end flow with dummy data
8. Set up monitoring and alerts
9. Train HR team on the system
10. Go live with pilot job posting
11. Monitor and iterate based on feedback

Document Version: 1.0
Last Updated: December 2025
Maintained By: HR Tech Team