Shashank Mishra — 7-Day AI Automation & Integration Sprint

Objective:

Transform the BHIV-HR-Platform into a fully AI-driven, N8N-integrated, automation-ready system connecting all three portals (Client, HR, Candidate) with real-world communication flows.

Sprint Outcomes

By the end of this 7-day sprint:

Candidate, HR, and Client portals are connected via APIs

AI agents (from BHIV Core) respond dynamically to candidate & recruiter actions

N8N automations handle WhatsApp, Email, Telegram, and (Vaani–Karthikeya) voice interactions

The platform can onboard → manage → offboard candidates with automated notifications and updates

Candidate-facing UI functional (in collaboration with Nikhil's frontend design)

Task Breakdown

Day 1–2: Candidate Portal Build + Backend Integration

- Create Candidate Portal APIs:
  - /candidate/register, /candidate/login
  - /candidate/profile (CRUD: education, experience, certificates, skills)
  - /candidate/applications (view applied jobs, status, tasks)
- Integrate Job Listings API (from HR portal)
- Connect to BHIV Core (Nisarg) for AI-based JD recommendations
- Save candidate actions/events to BHIV Bucket (Nipun)

Deliverable: Candidate portal backend ready + connected with HR and Client modules.

Day 3: Communication Layer Integration (N8N Setup)

- Deploy N8N workflows for:
  - Email notifications (job updates, task assignments)
  - WhatsApp updates (via Twilio / Meta API)

  - Telegram bot (status inquiries, interview reminders)

- Trigger automation events via API hooks:

  - When HR shortlists → WhatsApp/Email sent
  - When Client schedules → Telegram/Email sent
  - When Candidate submits feedback → HR notified

Deliverable: Working N8N instance with sample flow automations and webhook triggers from backend.

Day 4–5: AI Voice & Feedback Integration (Vaani + Karthikeya)

- Integrate Vaani (STT) and Karthikeya (TTS) modules:
  - Voice-based candidate interaction (e.g., “Confirm interview time”, “Provide feedback”)
  - Voice-to-text conversion → routed to BHIV Core for interpretation
- Create feedback loop:
  Candidate/HR voice feedback → sentiment → reward (to Ishan’s RL loop)
- Add audit logs for each voice interaction (timestamp, transcript, sentiment result)

Deliverable: Working voice-based interaction pipeline between candidate ↔ HR using Vaani/ Karthikeya.

Day 6: Cross-Portal AI Agent Synchronization

- Align agent events between portals:
  - When HR updates status → Client dashboard auto-refreshes
  - Candidate confirmation updates recruiter feed
  - RL feedback updates BHIV Core policy state
- Build agent-state synchronizer service (Node/Python microservice)
- Ensure real-time updates (Socket.IO or polling) across dashboards

Deliverable: Seamless 3-way data sync with AI agents handling background reasoning.

Day 7: Testing, Demo Prep & Documentation

- Conduct end-to-end testing:

    - Candidate applies → HR shortlists → Client schedules → Candidate feedback → RL updates

- Record automation demo (showing N8N triggers, AI voice, and dashboard sync)
- Write README.md covering:
    - APIs
    - N8N setup guide
    - Automation logic
    - Voice module usage
    - Integration architecture diagram

Deliverable: Fully documented, demo-ready AI-powered HR Platform.

Integration Alignment (No Overlap)

Module	Owner	Responsibility
Core Backend + Automation	Shashank	APIs, N8N, voice, sync
RL + Feedback Agent	Ishan	RL loop + reward learning
BHIV Core	Nisarg	Agent reasoning, MCP integration
BHIV Bucket	Nipun	Persistent data storage
Frontend	Nikhil/ Yash	UI/UX of all portals

Tech Stack Notes

- Backend: Node.js + Express + MongoDB (or Supabase)
- Automation: N8N (email, WhatsApp, Telegram, triggers)
- Voice: Vaani (STT) + Karthikeya (TTS)
- AI Integration: Python (Flask or FastAPI microservice from BHIV Core)
- Comms Layer: Socket.IO for live updates
- Frontend (with Nikhil): React/Next.js for UI implementation

Goal

Upgrade the BHIV HR Platform into a self-learning, AI-automated system using:

- Reinforcement Learning (RL) → adaptive match scoring
- N8N automation → email, WhatsApp, Telegram workflows
- MCPs (Model Context Protocols) → real-time AI decision integration
- Voice layer (STT/TTS) → for recruiter-candidate interactions

LEARNING KIT

1. RL + Feedback Loops (Core Learning)

Concepts to grasp:

- State, Action, Reward (SARSA / Q-learning)
- Human-in-the-loop RL
- Continuous learning loops (feedback-based reward updates)
- Integrating RL into production APIs

Recommended:

- “Reinforcement Learning for AI Agents – Sentdex”
  YouTube
- “Online RL Agents in Python” – CodeEmporium
  Search: “Online RL Python gym”
- “Reward Models in AI Systems” – OpenAI Cookbook
  Search: “OpenAI Reinforcement Learning feedback reward”

2. N8N Workflow Automations

Goal: Build no-code automation between BHIV HR backend and external channels.

Learn:

- Creating triggers (new job post, new candidate, feedback received)
- Connecting to Gmail/Outlook → send interview mails
- WhatsApp (Twilio or Meta Cloud API)
- Telegram bots (message + feedback capture)
- API call nodes for integrating RL agent

Recommended:

- “n8n Complete Workflow Automation Tutorial” – Code with Tomi
  YouTube
- n8n Docs: https://docs.n8n.io

3. MCP (Model Context Protocol)

Goal: Enable real-time coordination between AI models and services (OpenAI, local embeddings, RL loop).

Learn:

- MCP fundamentals: shared context between models
- Setting up local MCP server
- Connecting external AI services (LLMs, embeddings)
- Passing contextual RL data between components

Recommended:

- OpenAI MCP Guide
  🔗 https://github.com/modelcontextprotocol
- “How Model Context Protocol Works” – AssemblyAI / OpenAI Devs
  🔍 Search: “Model Context Protocol explained”

4. Voice Layer (STT + TTS Automation)

Goal: Integrate automated voice updates to candidates & HR teams

Stack:

- STT (Speech-to-Text): Vaani (Karthikeya), OpenAI Whisper / Vosk
- TTS (Text-to-Speech): Vaani (Karthikeya)

Recommended:

- “Build a Voice Bot with Whisper + OpenAI API” – freeCodeCamp
  YouTube
- Whisper API: https://platform.openai.com/docs/guides/speech

5. Integration Vision

Shashank’s platform should connect:

```
Recruiter Actions → RL Reward Update → N8N Trigger →
Candidate/Client Notification
```

RL learns from recruiter feedback

MCP syncs AI context between recruiter, candidate, and client portals

N8N pushes updates (email/WhatsApp/Telegram/Voice)