No-code/low-code software development course

"Turn complex processes into simple automated workflows. Move faster with flexible, scalable automation."

n8n

www.n8n.io

- ❑ Develop Workflows
- ❑ Perform Automations
- ❑ Execute Integrations
- ❑ Connect 1000+ online data sources

About Instructor:

Shahzad - Software Fractional Strategy Consultant - 25+ Years Of Experience In Software Industry - With Clients In Across North America And Europe, With Focus On:

- ❑ Cloud/SaaS- Software As Service
- ❑ Artificial Intelligence & Generative AI & LLM
- ❑ No-code/Low-code Software Development

About Institute:

NexSkill is Pakistan's leading IT training institute, providing the best IT training in Pakistan. NexSkill offers a variety of professional learning platforms to its students.

https://www.nexskill.com/

What Is n8n?

- n8n is an open-source, fair-code workflow automation platform used to connect apps, APIs, and data into automated workflows. [serveravatar.com], [digitalocean.com]
- It combines visual no-code workflow design with optional custom code, making it suitable for both non-technical users and developers. [go.lightnode.com], [hatchworks.com]
- n8n can be self-hosted or used via a managed cloud offering, giving users full control over data and infrastructure if needed. [docs.n8n.io], [arimetrics.com]

Comparison: n8n vs Zapier vs Make vs Airflow

n8n

Criteria	n8n	Zapier	Make (Integromat)	Apache Airflow
Tool Type	Open-source workflow automation	SaaS no-code automation	Visual low-code automation	Workflow orchestration framework
Primary Audience	Developers & technical teams	Non-technical business users	Power users & analysts	Data engineers & DevOps teams
Automation Style	Event-driven & API-based workflows	Event-based app automation	Visual scenario-based automation	Scheduled batch pipelines (DAGs)
Interface	Visual canvas + optional code	Simple linear UI	Visual flow editor	Code-first (Python)
Coding Support	JavaScript & Python supported	Very limited	Minimal scripting	Full Python programming required
Hosting Options	Cloud or fully self-hosted	Cloud-only	Cloud-only	Self-hosted
Data Control	Full control (self-hosting)	Limited	Limited	Full control
Integrations	1000+ native + any API	8,000+ app connectors	1,500+ connectors	Limited, requires custom code
Complex Logic Handling	Strong (loops, branching, retries)	Limited	Moderate	Very strong
Error Handling & Debugging	Visual step-by-step execution	Basic	Visual execution path	Logs & retries in code
Scalability	High (queue mode, workers)	Limited by plan	Moderate	Very high
Pricing Model	Free (self-hosted) / execution-based	Per task / action	Per operation	Free (infrastructure cost only)
AI & API Workflows	Excellent support	Limited	Moderate	Custom implementations
Best For	Flexible, production-grade automation	Simple business task automation	Medium-complex visual automations	Large-scale data pipelines & ETL

When to Choose Each Tool

n8n

✓ Choose n8n if

- ❑ You need **flexibility, custom logic, and API-first automation**
- ❑ Data privacy and **self-hosting** are important
- ❑ You want a balance between **visual workflows and real code**
- ❑ Common for DevOps, backend systems, and AI workflows

✓ Choose Zapier if

- ❑ You want **quick, simple automations** with no technical setup
- ❑ Automation logic is linear and low-complexity
- ❑ You rely heavily on **SaaS app connectors**

✓ Choose Make if

- ❑ You want a **visual tool with more logic than Zapier**
- ❑ You are comfortable with moderately complex scenarios
- ❑ You do not require self-hosting or deep backend control

✓ Choose Airflow if

- ❑ You are building **large-scale data pipelines or ETL jobs**
- ❑ You need **strict scheduling, dependency control, and retries**
- ❑ You are comfortable writing and maintaining **Python code**
- ❑ Common in data engineering and analytics platforms

Summary

- ❑ **Zapier** → Fast, simple, business-friendly
- ❑ **Make** → More visual power, moderate complexity
- ❑ **n8n** → Developer-friendly, flexible, self-hosted automation
- ❑ **Airflow** → Code-first, enterprise-grade data orchestration

How n8n Works

n8n

- Automation logic is built as **workflows**, which define a sequence of st eps executed automatically. [digitalocean.com] Each workflow consists of **nodes**, where each node performs a single action such as:
  - ✓ Listening for an event (trigger)
  - ✓ Calling an API
  - ✓ Transforming data
- Sending messages or updating systems. [arimetrics.com], [go.lightnod e.com]
- **Trigger nodes** start workflows based on schedules, webhooks, or man ual executions. [go.lightnode.com], [digitalocean.com]
- Data flows between nodes as **JSON**, enabling structured processing, b ranching, and looping. [serveravatar.com]
- n8n provides **visual execution logs**, allowing users to inspect node-le vel inputs and outputs for easy debugging. [go.lightnode.com]

Core Features

n8n

- **Visual workflow builder** using a drag-and-drop, node-based canvas. [it-learning-hub.com], [arimetrics.com]
- **Hybrid no-code / low-code model**, with support for JavaScript or Python when advanced logic is needed. [codecademy.com], [infralovers.com]
- **Hundreds of built-in integrations**, plus the ability to connect to any API via HTTP requests. [digitalocean.com], [it-learning-hub.com]
- **Self-hosting support** for privacy, compliance, and cost control. [arimetrics.com], [docs.n8n.io]
- **Advanced workflow logic**, including conditions, branching, batching, and error handling. [arimetrics.com]

Key Benefits of n8n

- **Full data ownership** when self-hosted, avoiding vendor lock-in. [arimetrics.com]
- **High flexibility and scalability**, from simple automations to enter prise-grade workflows. [hatchworks.com]
- **Cost-effective** compared to per-task pricing automation platforms. [go.lightnode.com]
- **Accessible for beginners, powerful for developers**, enabling a smooth learning curve. [serveravatar.com]

n8n Deployment Options

n8n

1. n8n Cloud

- ❑ Fully **managed cloud service** provided by n8n.
- ❑ No infrastructure or maintenance required.
- ❑ Quick setup, automatic updates, and backups.
- ❑ Suitable for teams that prefer **ease of use and faster onboarding**.
- ❑ Less control over infrastructure compared to self-hosting.

📌 Reference: https://docs.n8n.io/choose-n8n/cloud/

2. Local Installation (npm, Docker)

- ❑ Installed on a **local machine** for development or testing purposes.
- ❑ Supported methods:
  - ❑ **npm / npx** (Node.js environment)
  - ❑ **Docker** for containerized execution.
- ❑ Ideal for learning, prototyping, and experimenting with workflows.
- ❑ Not recommended for production without additional security and persistence setup.

📌 Reference: https://docs.n8n.io/hosting/installation/

3. Self-Hosting on Servers (Linux, Cloud VM)

- ❑ Deploy n8n on **own servers or cloud virtual machines** (AWS, Azure, GCP, etc.).
- ❑ Commonly uses **Docker or Docker Compose** for production setups.
- ❑ Provides **full control over data, security, scaling, and integrations**.
- ❑ Suitable for enterprises, regulated environments, and high-volume automation.
- ❑ Requires infrastructure management and monitoring.

📌 Reference: https://docs.n8n.io/hosting/

n8n Interface Walkthrough

n8n

1. Canvas and Node Editor

- ❑ Visual **drag-and-drop canvas** where workflows are designed.
- ❑ Nodes represent triggers, actions, logic, or integrations.
- ❑ Connections define **data flow and execution order**.
- ❑ Node editor allows configuration of parameters, credentials, and expressions.

2. Execution View

- ❑ Displays **real-time and past workflow executions**.
- ❑ Shows which nodes executed successfully or failed.
- ❑ Allows inspection of **input and output data** at each node.
- ❑ Helps debug workflows without re-running the entire flow.

3. Logs and Workflow History

- ❑ Maintains a record of **execution history** for each workflow.
- ❑ Logs include timestamps, status, and error details.
- ❑ Supports troubleshooting, auditing, and performance analysis.
- ❑ Useful for monitoring running workflows in production environments.

Node Types in n8n

n8n

a) Trigger Nodes

- Start a workflow when a specific **event or condition** occurs.
- Only **one trigger node** is required to initiate a workflow.
- Examples: Manual Trigger, Webhook Trigger, Schedule Trigger.

b) Core Nodes

- Provide **built-in logic and utility functions**.
- Used for data transformation and flow control.
- Examples: Set, IF, Switch, Merge, SplitInBatches.

c) Regular Nodes

- Perform **actions or integrations** after the workflow has started.
- Typically interact with external services or APIs.
- Examples: HTTP Request, Email, Slack, Database nodes.

Node Inputs, Outputs, and Connections

- **Inputs** receive data from the previous node in JSON format.
- **Outputs** pass processed data to the next connected node.
- **Connections** define the execution order and data flow.
- Each node executes once per incoming data item unless configured otherwise.
- Visual connections make complex workflows easier to understand and debug.

Trigger Nodes in n8n

n8n

1. Manual Trigger

- Used to **manually start** a workflow.
- Commonly used for testing, debugging, and development.
- Does not wait for an external event.

2. Webhook Trigger

- Starts a workflow when an **HTTP request** is received.
- Typically used to integrate with external applications or services.
- Supports real-time automation scenarios.
- Can handle different request methods (GET, POST, etc.).

3. Schedule (Cron) Trigger

- Starts workflows on a **time-based schedule**.
- Supports intervals such as minutes, hours, days, or custom cron expressions.
- Ideal for recurring tasks like reports, sync jobs, and data cleanup.

Data Flow Model in n8n

1. JSON Structure in n8n

n8n passes data between nodes using **JSON (JavaScript Object Notation)**.
Each node processes data as structured key–value pairs.
JSON ensures consistent, readable, and system-agnostic data exchange.
Example structure:

```
{
  "name": "John",
  "email": "john@example.com",
  "status": "active"
}
```

2. Items, Fields, and Expressions

Items:

Each workflow execution can contain **one or more items**.
Every item is processed independently through the workflow.

Fields:

Individual data points inside an item (e.g., *name*, *email*, *id*).

Expressions:

Used to dynamically reference and manipulate data.
Written using **double curly braces**: *{{ }}*.
Allow accessing values, transforming strings, dates, and numbers.
Example expression:

```
{{ $json.email }}
```

Data Flow Model in n8n

n8n

3. Understanding *$json*, *$node*, and *$input*

$json

Refers to the **current item's JSON data**.

Most commonly used in node expressions.

```
{{ $json.name }}
```

$node

Accesses data from **other nodes** in the workflow.

Useful when referencing earlier steps.

```
{{ $node["HTTP Request"].json.status }}
```

$input

Refers to the **incoming data** of the current node.

Useful in advanced transformations and code nodes.

```
{{ $input.first().json.id }}
```

✓ **Key Takeaway**

n8n's data model is **item-based and JSON-driven**, making workflows predictable, debuggable, and powerful when com
bined with expressions.

n8n – Core Nodes

1. Set Node

- Used to **create, modify, or remove fields** in the workflow data.
- Commonly applied to **prepare or clean data** before sending it to another node.
- Can set **static values** or use **expressions** for dynamic content.
- Helps reduce payload size by keeping only required fields.
- Does not call external services; operates only on data.

2. IF / Switch Nodes

IF Node

- Executes different paths based on **true/false conditions**.
- Useful for simple decision-making logic.
- Compares values using operators such as equals, contains, greater than, etc.

Switch Node

- Routes data to **multiple possible branches** based on a single field.
- Useful when handling more than two conditions.
- Improves readability over multiple IF nodes.

n8n – Core Nodes

3. Merge and SplitInBatches Nodes

Merge Node

- ❑ Combines data from **multiple branches** into one.
- ❑ Supports different merge strategies (append, merge by key, keep separate).
- ❑ Often used after parallel workflow paths.

SplitInBatches Node

- ❑ Breaks large datasets into **smaller batches**.
- ❑ Processes items sequentially instead of all at once.
- ❑ Useful for **rate-limited APIs** and large data operations.
- ❑ Improves performance and prevents execution timeouts.

✓ Key Takeaway

Core nodes control **data structure, flow logic, and performance**, forming the foundation of efficient n8n workflows.

n8n – Handling Errors Gracefully

n8n

1. Continue on Fail

- ❑ Allows a node to **fail without stopping the entire workflow**.
- ❑ When enabled, the workflow continues execution even if that node encounters an error.
- ❑ Useful when:
  - ✓ A failure is **non-critical** (e.g., optional API calls).
  - ✓ Processing large batches where some items may fail.
- ❑ Error details are still recorded in execution data for review.
- ❑ Helps build **resilient workflows** that do not break on minor issues.

2. Error Workflows

- ❑ Special workflows designed to **handle errors globally**.
- ❑ Trigger automatically when another workflow fails.
- ❑ Common uses include:
  - ✓ Capturing error messages and metadata.
  - ✓ Logging errors to databases or monitoring tools.
  - ✓ Performing cleanup or retry logic.
- ❑ Promotes **centralized error handling** across multiple workflows.

n8n – Handling Errors Gracefully

n8n

3. Notifications and Alerts

- ❑ Used to **inform teams when errors occur**.
- ❑ Can send alerts via:
  - ✓ Email
  - ✓ Slack, Microsoft Teams, or other messaging tools
- ❑ Usually integrated with error workflows for real-time notifications.
- ❑ Helps reduce downtime by enabling **faster detection and response**.
- ❑ Can be filtered to trigger alerts only for critical failures.

✓ Key Takeaway

Graceful error handling in n8n ensures workflows remain **stable, observable, and maintainable**, even when external sy stems fail.

Same Apps – Ecosystem as discussed in Zapier

n8n

SaaS App – Ecosystem:
https://github.com/ShahzadSarwar10/LOW-CODE-NO-CODE-B-1/tree/main/Zapier/Week3
[See above top 25 top SaaS App – Learning Material , study Material]
[Process and absorb – all basic information – about above 25 apps]

Top 500 SaaS App - category wise arranged:
https://github.com/ShahzadSarwar10/LOW-CODE-NO-CODE-B-1/blob/main/Zapier/Week1/Zapier_Top_500_Real-Apps.xlsx

Top SaaS – Multi step Zaps:
https://github.com/ShahzadSarwar10/LOW-CODE-NO-CODE-B-1/blob/main/Zapier/Week1/100_Complex_Multi_Step_Zaps-PracticeAreas.xlsx

Understand – "Revenue Operations"- What? Why? How? Which? Who? End to end processing
https://github.com/ShahzadSarwar10/LOW-CODE-NO-CODE-B-1/blob/main/Zapier/Week1/RevenueOperations-RevOps-TheFuture.pdf

Kindly read slowly and slowly.
Kindly read carefully and carefully.
All the best.
Thanks

n8n

www.n8n.io

- ❑ Develop Workflows
- ❑ Perform Automations
- ❑ Execute Integrations
- ❑ Connect 1000+ online data sources

Thanks for Your time