Assignment #5:

Practical Workflows in n8n

Behnam Mo

November 3, 2026

Build each workflow yourself, one node at a time, and run it as you go. You may ask an AI how a node works, but do not use n8n's AI Workflow Builder, or any other AI, to build or edit a workflow. In each workflow's settings, keep Save manual executions and Save successful production executions on. When a workflow is finished, open its Executions tab, scroll to the earliest execution, and open it, so the canvas shows the workflow as it was then. Paste a screenshot of that view under its part; it shows that you built the workflow step by step. Take it right away, because n8n deletes old executions. A part whose Executions shot is missing, or whose earliest execution already shows every node of the finished workflow, gets 0 for everything except its paragraph.

Wherever a workflow sends email, send it to your own address. Export each workflow as JSON, name each file after its part (Q1A, Q1B, Q2), and put the files in a zip with your document. Upload the zip to Canvas as `Assignment5_YourName.zip`.

Q1 [60 pt]. Two workflows you could put to work this semester.

A [30 pt]. Job leads every morning. You want an entry-level marketing job, and you want to see new openings before everyone else does. Build a workflow that runs every day at 8:00 a.m. and emails you the leads that suit you best.

- Put your CV in a Google Doc. It does not need to look nice; only an AI will read it. You do not need to share it: the Google Docs node signs in with your Google account, as the Gmail node did in class, and reads the document from its URL.
- Pick one entry-level marketing job, such as digital marketing coordinator, and write three different search phrases that would turn up new openings for it. Every morning,

the workflow searches all three with Brave Search and keeps the top 3 to 5 results of each.
- An AI gets the search results and the text of your CV. The CV goes into the AI's prompt, not into a tool. The AI picks the leads most likely to work out for you and says why.
- The workflow emails the AI's recommendation to you.

There are two ways to run the searches, and both are correct:

1. The workflow searches, combines the results into one item, and passes that item to the AI.
2. The AI gets Brave Search as a tool and the three phrases in its prompt, and searches on its own.

Set the Schedule Trigger to every day at 8:00, set the workflow's time zone to America/Chicago in its settings, and publish the workflow; a schedule never fires in a workflow that is not published. Publish it by November 12, so that it runs on at least three mornings before the deadline.

Tip: If the workflow does the searching, put the three phrases in a data table, one per row, and the Brave Search node runs once for each. Three search nodes side by side, all joined into one node, make every node after them run three times, so you would get three emails.

In your document, paste:

1. Your three search phrases, and a screenshot of the canvas after one morning's run, opened from the Executions tab, with every node green. **[6 pt]** for searches that return 3 to 5 results for each phrase, and **[3 pt]** for your CV reaching the AI's prompt.
2. A screenshot of the Executions tab with at least three runs at 8:00 a.m. on three different days. **[9 pt]**
3. A screenshot of one morning's email. **[6 pt]** if it names specific openings from that morning's results, with links, and says why each one fits your CV.
4. One paragraph: which of the two ways costs less to run, and why? **[6 pt]**

B [30 pt]. Sublease your room. You are leaving for a summer internship and need someone to take over your room near campus. Build a form for people who want the room, and have the workflow answer each of them.

- The form asks for the person's name, email, monthly budget, move-in date, and a few words about themselves.
- Every submission becomes a row in a data table.
- An IF node decides who fits: the budget is at least your rent, and the move-in date is within the next 30 days. Choose your own rent, and describe your place, real or imagined, to the AI.
- A person who fits gets a personal reply, written by an AI from what they wrote and from your room's details. Everyone else gets the same fixed email, which you write once.

Tip: In the IF node, compare the move-in date with the expression `$now.plus({ days: 30 })`, which is the date 30 days from now.

Test it with three submissions, each with your own email: one person who fits, one whose budget is below your rent, and one who wants to move in more than 30 days from now. In your document, paste:

1. A screenshot of the canvas after each run. **[9 pt]**, 3 for each run that takes the right branch.
2. A screenshot of the data table with the three rows. **[6 pt]**
3. The email each test received. **[6 pt]** for a personal reply to the person who fits, and **[3 pt]** for the same fixed email to the other two.
4. One paragraph: why are the budget and date rules an IF node, rather than a question you ask the AI? **[6 pt]**

Q2 [40 pt]. Pearlwave Tea, a boba shop near campus, wants a Find my drink form on its Instagram page, so that customers can get a recommendation at any hour. Its menu is in pearlwave-menu.csv: 25 drinks with price, allergens, caffeine, and whether each is in stock today. Import it into a new data table.

The form asks for the customer's name, email, budget, allergy (a dropdown with No allergies, Dairy, Nuts, Gluten, and Soy), sweetness, and what they are in the mood for. Then the workflow:

1. Reads every row of the menu table.
2. Filters the rows: the price is at most the budget, the allergens do not contain the

customer's allergy, and the drink is in stock.

3. If no drink passes, emails the customer a fixed message saying nothing fits yet.
4. Otherwise, uses Edit Fields to keep only what the AI needs, aggregates the drinks into one list, and asks an AI to recommend up to three drinks, only from that list, for the customer's sweetness and mood.
5. Emails the recommendation to the customer.

Tip: In the Filter node's settings, turn on Always Output Data. Without it, a filter that keeps no drink ends the run, and the IF never runs.

Run three tests, each with your own email:

- Test 1: budget $7, no allergies, 50% sweet, mood "something fruity and refreshing".
- Test 2: budget $5.50, dairy, 25% sweet, mood "I need coffee".
- Test 3: budget $3, nuts, 50% sweet, mood "anything cheap".

In your document, paste:

1. A screenshot of the data table with all 25 rows. **[4 pt]**
2. For each test, a screenshot of the canvas after the run **[4 pt]** and the email it sent **[4 pt]**.
3. One paragraph: your job-lead workflow serves only you, so Claude Code could run it for you instead. Why does Pearlwave need this one in n8n, and why must the allergy check be a Filter rule rather than part of the AI's judgment? **[12 pt]**