### **🔧 Tech Stack Overview**

#### **1. Frontend (Optional, but great for UI-based workflows)**

* **Framework**: React.js (for building a modern, responsive UI)
* **State Management**: Redux or Zustand
* **Design**: Tailwind CSS or Material UI

#### **2. Backend (Core Focus Area)**

* **Language**: Java (Spring Boot)
* **Frameworks**:  
  + Spring Boot (main REST API framework)
  + Spring Security (auth and permissions)
  + Spring Scheduler (for timed jobs/workflows)
* **Data Handling**:  
  + Jackson (for JSON serialization)
  + Feign Client or WebClient (for calling external REST APIs)

#### **3. API Integration Layer**

* **REST Clients**: Custom-built for each service (e.g., Gmail, Google Drive, Trello)
* **Authentication**: OAuth 2.0 (for most third-party APIs)
* **Webhook Receiver**: Endpoints to trigger flows (like Zapier's triggers)

#### **4. Workflow Engine**

* **Task Orchestration**:  
  + Custom Java-based rule engine or lightweight BPMN engine (e.g., Camunda or Zeebe)
* **Queueing**:  
  + Apache Kafka or RabbitMQ (for decoupling workflow steps)

#### **5. Cloud & DevOps**

* **Cloud Provider**: AWS or GCP (choose based on familiarity or resume target)
* **Services**:  
  + Compute: AWS EC2 / GCP Compute Engine or App Engine
  + Storage: S3 / GCP Cloud Storage
  + Secrets Management: AWS Secrets Manager or GCP Secret Manager
* **Containerization**: Docker
* **CI/CD**: GitHub Actions or Jenkins

#### **6. Database**

* **Primary DB**: PostgreSQL or MongoDB (for storing workflows and logs)
* **In-memory Cache**: Redis (for fast session/state handling)

#### **7. Scripting Support**

* For extensibility, support scripting in:  
  + Node.js (via GraalVM or microservices)
  + Bash or Python (for external integrations or custom logic modules)

### **🗺️ Suggested Module Breakdown**

| **Module Name** | **Description** |
| --- | --- |
| **User Auth & Mgmt** | Sign-up/login, API keys, OAuth token management |
| **Workflow Designer** | UI or JSON-based flow creation (if UI included) |
| **Trigger Module** | Handles incoming webhooks/events |
| **Action Executor** | Calls relevant APIs based on workflow steps |
| **Scheduler & Queue** | Executes tasks based on time or order |
| **Logging & Monitoring** | Logs execution status, errors, retries |
| **Admin Panel (Optional)** | For managing workflows, users, usage metrics |