In "JobRequisition" table, in addition to the JR related fields, there is a field named "jr_status" for specifying the status of the JR entry during the various stages of Approval process.

When a JR is submitted, an Approval workflow should trigger.

Approval flow procedures:-

The Job Requisition(JR) submission and Approval workflow is depicted in the attached file(refer ApprovalFlow.jpg). From the attached file it is clear that either Hiring Manager(HM) or DU Head can raise a JR. Also the detailed steps need to be performed once the JR is submitted through the application is mentioned below. Please create the DB tables for the approval workflow in a configurable manner. Also please create the UI design for the approval workflow configuration if it is configurable. In this approval workflow configuration screen, you need to specify a column "Waiting period for Approval" which tells within how many days an approver should approve/reject a JR assigned to him. If any approver fails to approve his allocated JR within this time period, a reminder mail should be sent to him(refer row 4 of "Email Templates.pdf" for the mail format)

When a JR is submitted, first the application should check whether the submitter is Hiring Manager(HM) or DU Head by checking the field named "hiring_manager_id" in "JobRequisition" table. By checking the master table "User", we can find out whether the submitter is Hiring Manager(HM) or DU Head (Refer "role" field in "User" table).

If the submitter is Hiring Manager, the below 3 steps should be performed

1. "jr_status" column in "JobRequisition" table should be changed to "Pending DU Head Approval"

2. Find out the DUHead: Retrieve the "department_id" mentioned in "JobRequisition" table. Then find out the DUHead corresponding to this Department from the "Department" master table.

3. Send approval request mail to DU Head with JR details and a link to the Recruitment portal(refer SI No: 1 of "Email Templates.pdf" for the mail format). On clicking this link, the approver(DU Head) can login to the portal so that he can view the specified JR in the Dashboard and can Approve/Reject it by clicking either the "Approve" or "Reject" buttons displayed against the specified JR.

But If the submitter is DU Head, the below 3 steps should be performed

1. "jr_status" column in "JobRequisition" table should be changed to "Pending CDO Approval"

2. Find out the CDO: Retrieve the "department_id" mentioned in "JobRequisition" table. Then find out the CDO corresponding to this Department from the "Department" master table.

3. Send approval request mail to CDO with JR details and a link to the Recruitment portal(refer Sl No: 1 of "Email Templates.pdf" for the mail format). On clicking this link, the approver(CDO) can login to the portal so that he can view the specified JR in the Dashboard and can Approve/Reject it by clicking either the "Approve" or "Reject" buttons displayed against the specified JR.

When DU Head approved, the below 4 steps should be performed

1. "jr_status" column in "JobRequisition" table should be changed to "Pending CDO Approval"

2. Send notification mail to Submitter (Hiring Manager) intimating the DU Head approval (refer Sl No: 2 of "Email Templates.pdf" for the mail format)

3. Find out the CDO: Retrieve the "department_id" mentioned in "JobRequisition" table. Then find out the CDO corresponding to this Department from the "Department" master table.

4. Send approval request mail to CDO with JR details and a link to the Recruitment portal(refer Sl No: 1 of "Email Templates.pdf" for the mail format). On clicking this link, the approver(CDO) can login to the portal so that he can view the specified JR in the Dashboard and can Approve/Reject it by clicking either the "Approve" or "Reject" buttons displayed against the specified JR.

When CDO approved, the below 3 steps should be performed

1. "jr_status" column in "JobRequisition" table should be changed to "Pending COO Approval"

2. Send notification mail to Submitter(Hiring Manager/DU Head) intimating the CDO's approval (refer Sl No: 2 of "Email Templates.pdf" for the mail format)

3. The EmailId of COO is mentioned in the Application config file. Send approval request mail to COO with JR details and a link to the Recruitment portal (refer Sl No: 1 of "Email Templates.pdf" for the mail format). On clicking this link, the approver(COO) can login to the portal so that he can view the specified JR in the Dashboard and can

Approve/Reject it by clicking either the "Approve" or "Reject" buttons displayed against the specified JR.

When COO approved, the below 3 steps should be performed

1. "jr_status" column in "JobRequisition" table should be changed to "Approved"

2. Send notification mail to Submitter(Hiring Manager/DU Head) intimating the COO's approval (refer SI No: 2 of "Email Templates.pdf" for the mail format)

3. JR is assigned to a Recruiter Lead: Retrieve the "department_id" mentioned in "JobRequisition" table. Then find out the "recruiter_lead" corresponding to this Department from the "Department" master table. Update "recruiter_lead_id" column of " JobRequisition" table with this "recruiter_lead" value.

4. Send notification mail to Recruiter Lead mentioning that he is assigned as the Recruiter Lead for this JR (refer SI No: 6 of "Email Templates.pdf" for the mail format)

5. Send notification mail to Submitter(Hiring Manager/DU Head) mentioning that the JR is assigned to a Recruiter Lead (refer SI No: 5 of "Email Templates.pdf" for the mail format)

6. When Recruiter Lead received this mail, he can then login to the portal and can allocate a Recruiter POC for this particular JR. The flow for allocating this JR to a RecruiterPOC by the RecruiterLead will be explained later.

During this approval process, if any of the approver rejects this JR due to any reasons, the JR needs to be revised. The below mentioned steps are performed.

1. "jr_status" column in "JobRequisition" table should be changed to "Rejected"

2. A notification mail is sent to the Submitter intimating the JR rejection and its reason (refer SI No: 3 of "Email Templates.pdf" for the mail format)

3. The submitter then revise this JR by logging into the Portal. In the Dashboard, he can see that rejected JR with a Button with caption "Revise". On clicking this Revise button, he should be able to edit this JR and can make the necessary changes. On clicking the Submit Requisition button the approval process is initiated again as per the above specified hierarchy. Please note that on clicking the Submit Requisition button, there is no need to do the JR Number generation step as this JR already has a JR Number associated with it. The approval process needs that same JR number for the remaining processes. Thereafter in the dashboard page, there is no need to show the "Revise" button against this JR as it is not currently in the "Rejected" status.

Once the approval process is completed, a Job Description(JD) needs to be created based on the submitted JR. JD creation actually means creating a DB table named "JobDescription" whose values should be populated from the "JobRequisition" table itself. "JobDescription" table is actually a subset of the "JobRequisition" table.

The "JobDescription" table fields:-

Id (primary key), jr_id, job_requisition_id(foreign key (JobRequisition(id)), primary_skills, secondary_skills, nice_to_have_skills, qualifications, certifications, specific_qualification, total_experience_min, total_experience_max, relevant_experience_min, relevant_experience_max, number_of_positions, expected_date_of_onboarding_start, expected_date_of_onboarding_end, work_locations, work_shift_id, shift_time, job_purpose, primary_duties, good_to_have_duties, job_specification, is_active, created_at, updated_at, submitted_by, work_arrangement, requested_date, department_id, hiring_manager_id, recruiter_lead_id, recruiter_poc_id

Please create this table in NeonDB. The field datatypes and constraints should be the same as that of the corresponding fields in the "JobRequisition" table. So when the approval process is completed, "JobDescription" table should also be filled with the relevant data.