



VIT AE HACKATHON

AUGUST 2023

This doc will have basic requirements for the assignment task. If you are familiar with any tools or libraries, feel free to use that. You don't have any constraints on languages, libraries, or frameworks. Your focus should be on building a product that fits the requirement.

REQUIREMENT

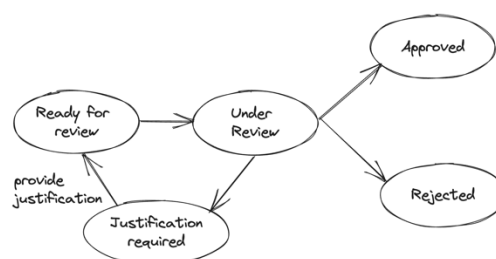
Develop an asynchronous approval system that enables users to submit requests for approval and allows designated approvers to review, approve, or reject those requests. This system can be useful in various scenarios, such as expense reimbursement, leave requests, and more.

User Roles:

1. Requesters
2. Approvers
3. Administrators

Project Flow:

1. Assume a different set of users already available in the system with required roles. No signups required
2. Login in as administrator and create some workflow with the required set of approvers and approval type.
3. Login in as requester and initiate a request with required details.
4. Once a request is initiated, send notification to the list of approvers.
5. Login in as approver, view pending requests, review it and then either approve or reject or move it to justification required step.
6. The requester can add more details and re-submit the application, if justification is required.
7. Once the request is moved to justification required or approved or rejected, send notification to the requester.



PAGES

Administrator View:

1. Can create new workflow and set the list of approvers.
2. A single workflow can have multiple approval types. Administrators can select the type when creating a workflow.
 - Everyone should approve.
 - At least two should approve.
 - Anyone can approve.
3. A dashboard to display the below.
 - Total number of pending workflows - (in number)
 - Total approved this month / this week / today - (in number)
 - Total rejected this month / this week / today - (in number)
4. Audit page where they can view the history of workflows approved/rejected.

Requester View:

1. Requester can request for a particular workflow (reimbursement/leave/etc)
 - a. Select the workflow type.
 - b. Add description.
 - c. Add attachments if any.
 - d. Submit the workflow.
2. View the list of active, approved, rejected requests which they have initiated.

Approver View:

1. Approver can view list of requests pending in his plate sorted based on timestamp.
2. Can click on a request, review it and either approve or reject.

KEY FEATURES

User Roles and Permissions:

- Define different user roles (requesters, approvers, administrators) with specific permissions.
- Allow administrators to manage user roles and access levels.

Workflow Creation:

- Allow administrators to create customizable approval workflows based on request types.
- Specify the sequence of approvers and conditions for approval/rejection.

Request Submission:

- Users can submit requests with relevant details and supporting documents/attachments.
- Different request types (e.g., expense request, leave request) can be defined.

Approver Assignment:

- Automatically route requests to the appropriate approver based on the defined workflow.

Approval Dashboard:

- Approvers have a dashboard displaying pending requests requiring their attention.
- Notifications and reminders for pending approvals.

Approval Process:

- Approvers can review request details, view attachments, and make informed decisions.
- Ability to add comments and notes during the approval process.

Approval History:

- Maintain a transparent record of approval decisions and timestamps.

Notifications and Alerts:

- Send email notifications to requesters and approvers at different stages of the approval process.

API ENDPOINTS

- Please implement all the necessary API endpoints that fits the requirement.

EXPECTATIONS

1. Clean professional level Code.
2. More weightages would be given to the functionalities than UI. Having a better UI will be a bonus.
3. HTML and CSS can be taken from the internet. Feel free to use any frameworks to polish the UI.
4. Basic requirements should be completed. Additional features/enhancements can be discussed with the us. **We highly encourage you to reach out to us. This is a collaborative session with Motorq team.**
5. Completing the project is more important than making it more perfect. It's okay if you don't include every detail, as long as it works well.
6. **We'll add a bar raiser requirement once you complete the entire implementation.** Keep up the good work!

GUIDELINES FOR SUBMISSION

Please use one of the below methods:

- Create a .zip file of all application related files and send an email (campus-hiring@motorq.co) with name and Roll number.
- Upload code to public GitHub repo and email us the link.
- Ensure that a README file is present in the base folder describing steps for any local steps required.
- **Important: Record the working project and add the link of the video in README file and the email (without demo, project will not be reviewed).**