

Before starting the fixes, please follow these steps: 1. First, pull the latest changes from the **Employer** branch. 2. Merge the **Job Seeker** branch into the Employer branch. 3. Resolve any merge conflicts carefully and ensure nothing breaks. 4. Verify that all existing functionality is working correctly after the merge. Once the merge is complete, please resolve all the issues/bugs I reported, including:

1. Bug – Forgot Password Email Not Working

<http://localhost:3000/forgot-password>

The **Forgot Password** functionality is currently not working correctly.

Issue

I tested the Forgot Password flow using a registered Gmail address, but **I did not receive any reset password email** in my Inbox or Spam folder.

Also, I noticed that **no API request is being triggered** when I click the **Send Reset Link** button. There is **no network call visible in the browser's Network tab**.

Please Check

- Verify why the Forgot Password API is not being called.
- Ensure the frontend is triggering the correct API request.
- Verify the API endpoint integration.
- Check whether the reset password email is being triggered after a successful API response.
- Verify the SMTP/email configuration.
- Ensure the correct email template is being used.
- Confirm that the reset link/token is being generated correctly.
- Check for any frontend or backend errors preventing the request.

Expected Behavior

1. User clicks **Forgot Password**.
2. Enters a registered email address.
3. Clicking **Send Reset Link** should trigger the Forgot Password API (visible in the Network tab).
4. On success, a reset password email should be sent to the registered email address.
5. The email should contain a valid reset password link.
6. Clicking the link should open the Reset Password page where the user can create a new password.

Please test the complete flow end-to-end and fix the issue so both the API call and email delivery work correctly.

2. ### Bug – Replace Dummy reCAPTCHA with Production reCAPTCHA

The login page is currently using a **dummy/test reCAPTCHA**.

As shown in the screenshot, it displays:

> "This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this."

This should **not** be visible in the production application.

Required Fix

- Remove the current dummy/test reCAPTCHA.
- Integrate the **real Google reCAPTCHA** (production version).
- Use valid Site Key and Secret Key from Google reCAPTCHA.
- Ensure reCAPTCHA verification is performed on both the frontend and backend.
- The warning message should no longer appear.
- Login should only be allowed after successful reCAPTCHA verification.

Expected Behavior

1. The user opens the Login page.
2. A real Google reCAPTCHA widget is displayed.
3. The user completes the verification.
4. The reCAPTCHA token is validated on the backend.
5. Login proceeds only if the verification is successful.

Please replace the dummy implementation with a fully functional Google reCAPTCHA integration.

Welcome Back!

Login to your account and continue

Role *

Select your role



Email Address *

Enter your email

Password *

Enter your password



☐ Remember me

[Forgot Password?](#)

Captcha *

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.



I'm not a robot



reCAPTCHA

→] Login

Don't have an account? [Register Now](#)

3. <http://localhost:3000/employer/my-profile>

Employer Profile – Make My Profile Fully Dynamic

The **Employer → My Profile** page is still using **static/mock data**. It needs to be fully integrated with the backend so all information is displayed dynamically.

Requirements

- Remove all hardcoded/static data from this page.
- Fetch all profile information from the backend APIs.
- Every section should display real data returned by the backend.
- If any required field/API is missing, please add it on the backend as well (except the Account Details section mentioned below).

Sections to Make Dynamic

- Company Profile
 - Company Logo
 - Company Name
 - Verification Status
 - Industry
 - Company Size
 - Website
 - Location
- Employer Profile
 - Profile Image
 - Name
 - Designation
 - Email
 - Phone Number
 - Joined Date
 - Primary Contact Badge
- Company Details
 - Company Type
 - Founded Year
 - Company Size
 - Industry
 - About Company
 - Website
- Contact Information
 - Phone Number

- Email Address
- Support Email
- Working Hours

- Address
 - Address
 - City
 - State
 - Country
 - Zip Code

- Account Plan
 - Current Plan
 - Plan Status
 - Plan Expiry
 - Remaining Credits (if applicable)

- Social Profiles
 - LinkedIn
 - Facebook
 - Twitter/X
 - Instagram
 - YouTube
 - Website
 - Any other social links available

Backend Changes

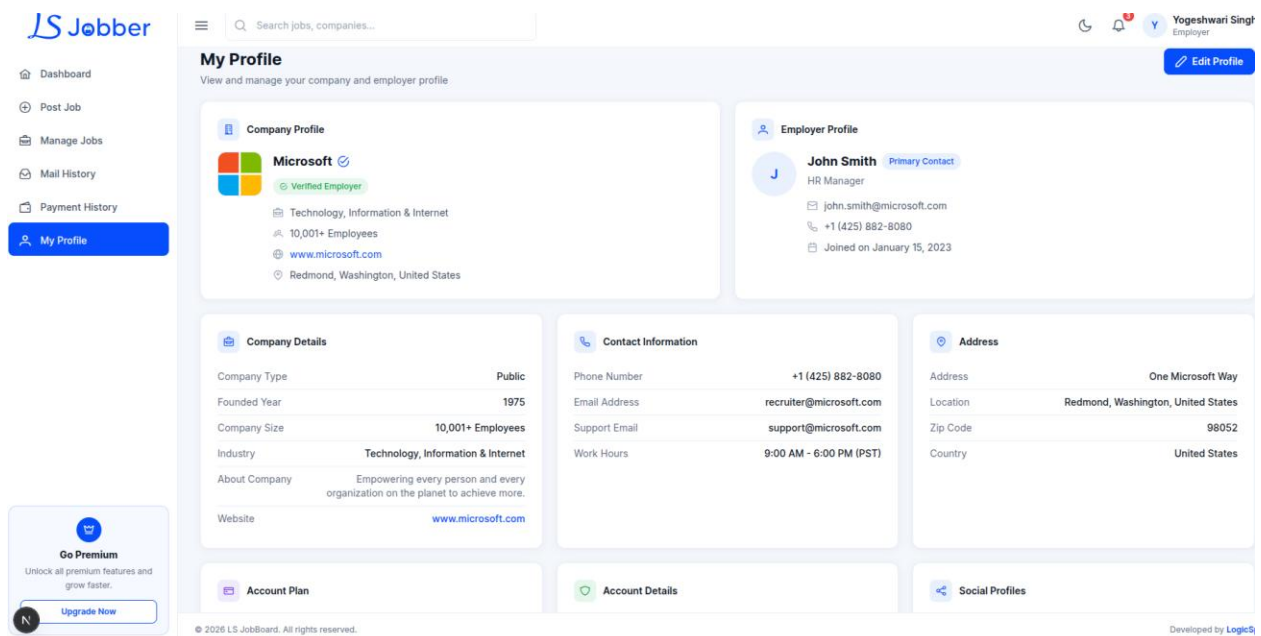
If any of the above fields are not currently available in the backend response, please add them so the frontend can render everything dynamically.

Exception

****Do NOT work on the Account Details / Account Number section right now.**** We will implement that at the end.

Expected Result

- No dummy/static values should remain.
- Every value should come from the backend.
- If a value is unavailable, show "-" or hide the field gracefully instead of displaying dummy data.
- Keep the current UI/UX exactly the same. Only replace the static content with dynamic backend data without breaking the existing design.



4. I noticed that the **Upgrade Plan cards are not being displayed on the Employer dashboard.**

Could you please confirm whether these plans will be fetched dynamically from the **Admin side**?

Also, please let me know:

- Will the subscription plans come from the Admin → Subscription Plans module?
- Are the plan cards currently missing because the backend integration is not completed yet?
- Is there any API pending for fetching the available plans?
- If not, is there another reason why the Upgrade Plan section is empty?

Please confirm how this section is intended to work so I can verify it properly during testing.

5. #### UI Enhancement – Notification Bell Animation

Please add a subtle animation to the **Notification Bell** icon in the header.

Requirements

- Add a smooth **ring/shake animation** so the bell appears to be gently ringing.
- The animation should feel modern and professional—not too fast or distracting.
- It should rotate slightly left and right from the top, similar to a real notification bell.
- Keep the notification badge/count fixed in place while the bell animates.
- The animation should be smooth and lightweight.

Animation Behavior

- The bell should animate every **5–8 seconds** when there are unread notifications.
- On hover, slightly scale the bell (e.g., `scale(1.05)`).
- Clicking the bell should not interrupt or break the animation.
- Respect the user's `prefers-reduced-motion` setting by disabling the animation when reduced motion is enabled.

Important

- Do not change the existing UI or layout.
- Only add the bell animation.
- Ensure the animation works consistently across all three dashboards (Admin, Employer, and Job Seeker).

6. ### UI Update – Job Details Page

<http://localhost:3000/jobs/549>

Please remove the following fields from the **Job Details** page, as they are no longer required:

- Industry
- Headquarters
- Contact Person
- Website

Requirements

- Remove these fields completely from the Job Details page UI.
- Do not leave any empty space or broken alignment after removing them.
- Adjust the layout so the remaining information is displayed properly.
- Do not affect any other sections or functionality on the page.

Please keep the rest of the Job Details page exactly as it is.

7. ### Refactoring – Create a Common Dropdown CSS Component

I'm providing the CSS for the dropdown design.

Please make this a **common reusable dropdown style/component** instead of defining the same CSS in multiple places.

Requirements

- Create a single common dropdown CSS/component using the provided design.
- Apply this common style everywhere the same dropdown UI is used.
- Remove all duplicate/repeated dropdown CSS from individual pages/components.
- Ensure every dropdown across the application uses the same common styling.
- The UI should remain exactly the same after refactoring.
- Do not change any functionality—this task is only for code optimization and maintainability.

Important

- Check all three dashboards (Admin, Employer, and Job Seeker).
- Replace any repeated dropdown CSS with the common reusable class/component.
- Do not break the existing design or functionality.
- Use the common dropdown style consistently throughout the project.

```
const dropdownPt = {
  root: {
    className:
      "flex h-11 w-full items-center justify-between rounded-
field border border-border bg-surface px-3",
  },
  input: { className: "flex-1 text-sm text-heading" },
  trigger: { className: "text-subtle" },
  panel: {
    className:
      "z-50 mt-1 rounded-card border border-border bg-surface p-1
shadow-lg",
  },
  // --- SEARCH / FILTER SECTIONS FIXED HERE ---
  header: {
    className:
      "p-2 border-b border-border bg-surface flex items-center
justify-between",
  },
  filterContainer: {
    className: "relative w-full flex items-center",
  },
  filterInput: {
    className:
```

```

        "w-full h-9 pl-8 pr-3 text-sm text-heading border border-
border rounded-md bg-surface focus:outline-none focus:border-
primary",
    },
    filterIcon: {
        className: "absolute left-2.5 text-xs text-subtle pointer-
events-none",
    },
    // -----
    wrapper: { className: "max-h-60 overflow-auto" },
    list: { className: "m-0 list-none p-0" },
    item: {
        className:
            "cursor-pointer rounded-md px-3 py-2 text-sm text-body
hover:bg-primary-soft hover:text-primary",
    },
    emptyMessage: {
        className: "p-3 text-sm text-subtle text-center",
    },
};

```

8.### Employer Dashboard – Dashboard State After First Job

After the employer posts their **first job**, the Empty State should no longer be displayed.

Instead, the Employer Dashboard should automatically switch to the **normal dashboard layout** (same as the UI shown in the reference screenshot).

Requirements

- Once at least **one job** exists, show the standard Employer Dashboard.
- Display all dashboard widgets and analytics, including:
 - Statistics cards (Active Jobs, Total Applications, Shortlisted Candidates, etc.)
 - Dashboard graphs/charts
 - Recently Added Jobs
 - Recent Applications
 - Latest activities
 - Any other dashboard widgets

Dynamic Data

All dashboard data should come from the backend, including:

- Active Jobs count
- Total Applications
- Shortlisted Candidates
- Profile Views (if available)
- Recently Added Jobs
- Recent Applications
- Graph/analytics data

Do **not** use any static or dummy data.

Dashboard Logic

- **No jobs posted** → Show the Empty State dashboard.
- **One or more jobs posted** → Automatically display the normal Employer Dashboard with real analytics and recent data.

Important

- Keep the existing dashboard UI exactly as it is (same layout as the provided design).
- Only make the data dynamic.
- Do not create a separate dashboard page.
- Ensure the dashboard updates automatically after the employer posts their first job.



Welcome to Your Dashboard!

You're all set! Let's get started by posting your first job and connect with the right candidates.

[+ Post Your First Job](#)[📘 Learn how it works](#)

Why Post a Job?



Reach Top Talent

Connect with qualified candidates actively looking for opportunities.



Quick & Easy

Post a job in minutes and start receiving applications.



Manage with Ease

Track applications and manage your jobs all in one place.



Secure & Reliable

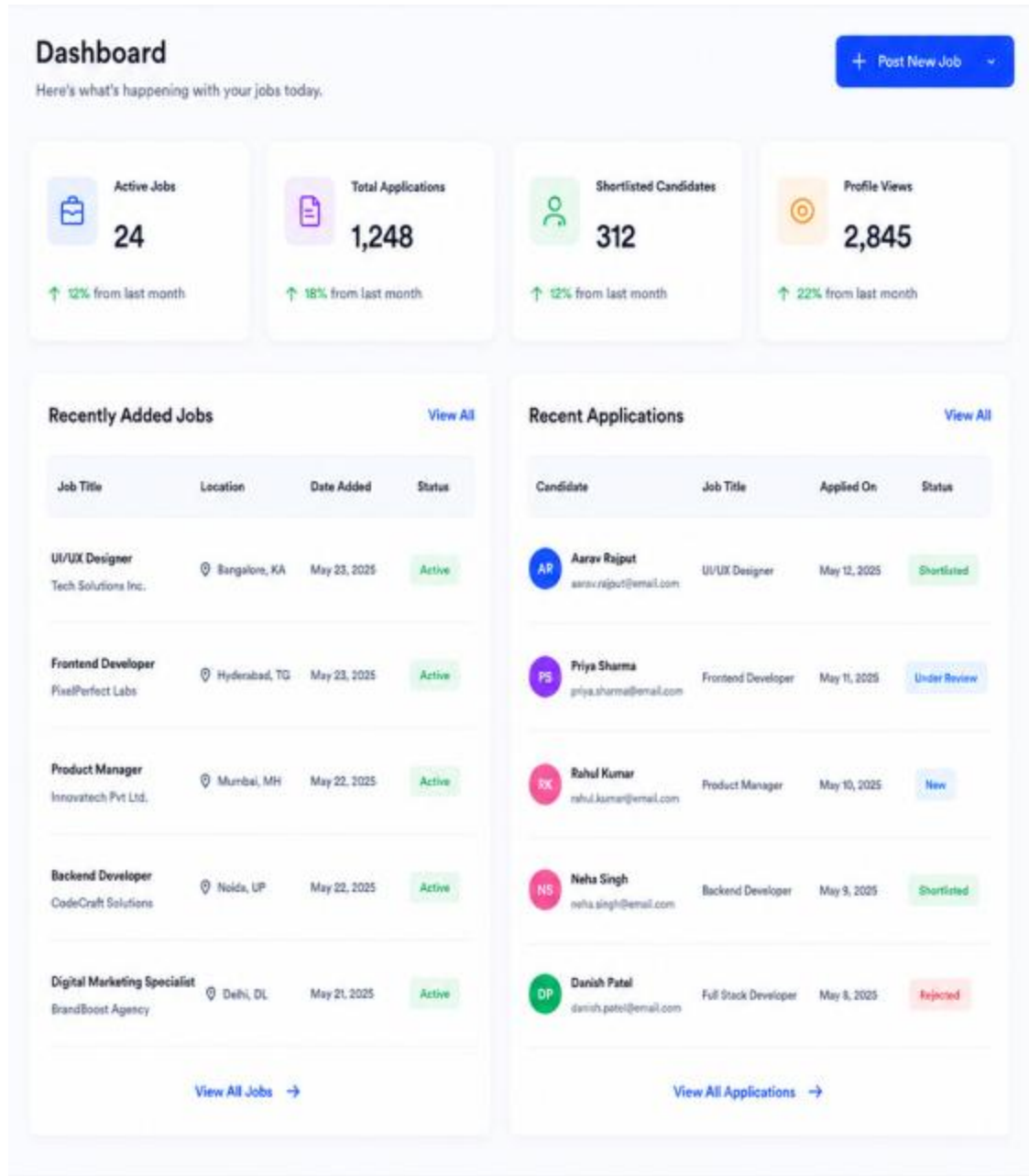
Your data and transactions are safe with us.



Go Premium

Unlock all premium features grow faster.

[Upgrade Now](#)



9. Mail History – Statistics Cards Fix

Currently, when there is **no mail history data**, the top statistics are still showing dummy values:

- This Month: 48
- This Week: 12
- Today: 3

This is incorrect.

Required Fix

Make these statistics fully dynamic.

If there is ****no email history****, then all counters should display ****0****.

Expected behavior:

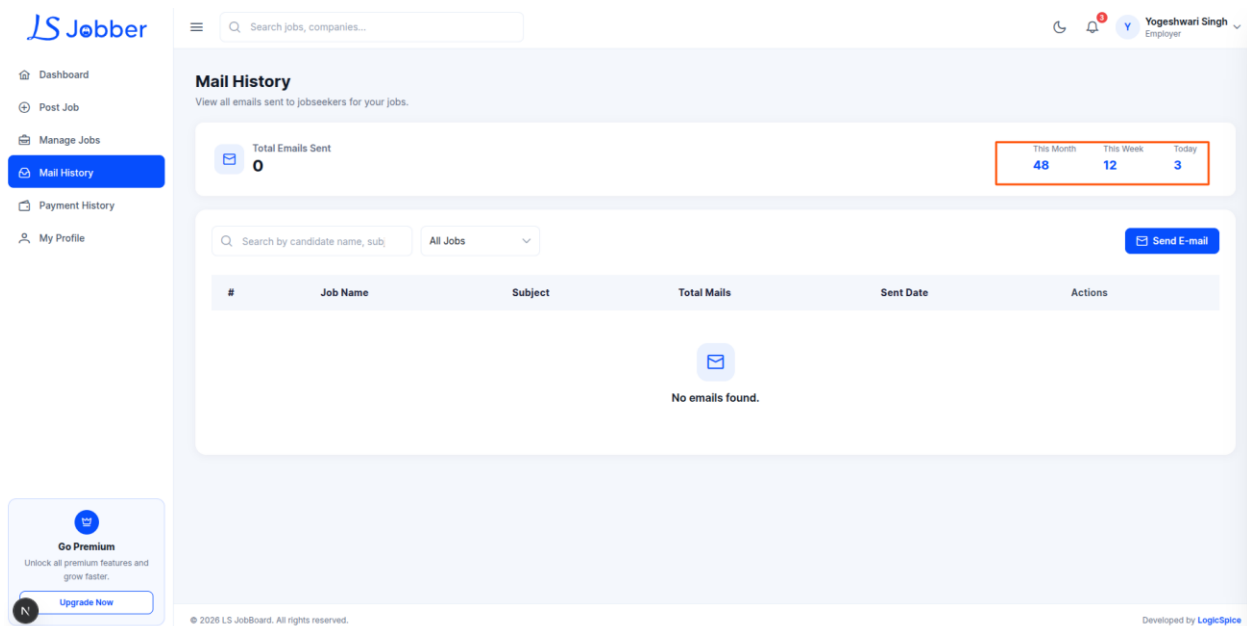
- Total Emails Sent → 0
- This Month → 0
- This Week → 0
- Today → 0

These values should be calculated from the actual Mail History records, not from static or dummy data.

Logic

- No mail records → Show 0 for every statistic.
- Once emails are sent, automatically update:
 - Total Emails Sent
 - This Month
 - This Week
 - Today

Remove all hardcoded numbers and bind these counters to the real backend/API response.



10. Mail History → Send E-mail popup has a validation issue.

Issue:

Even after entering a valid email address in the "To" field, I still get the validation error:

"Add at least one recipient."

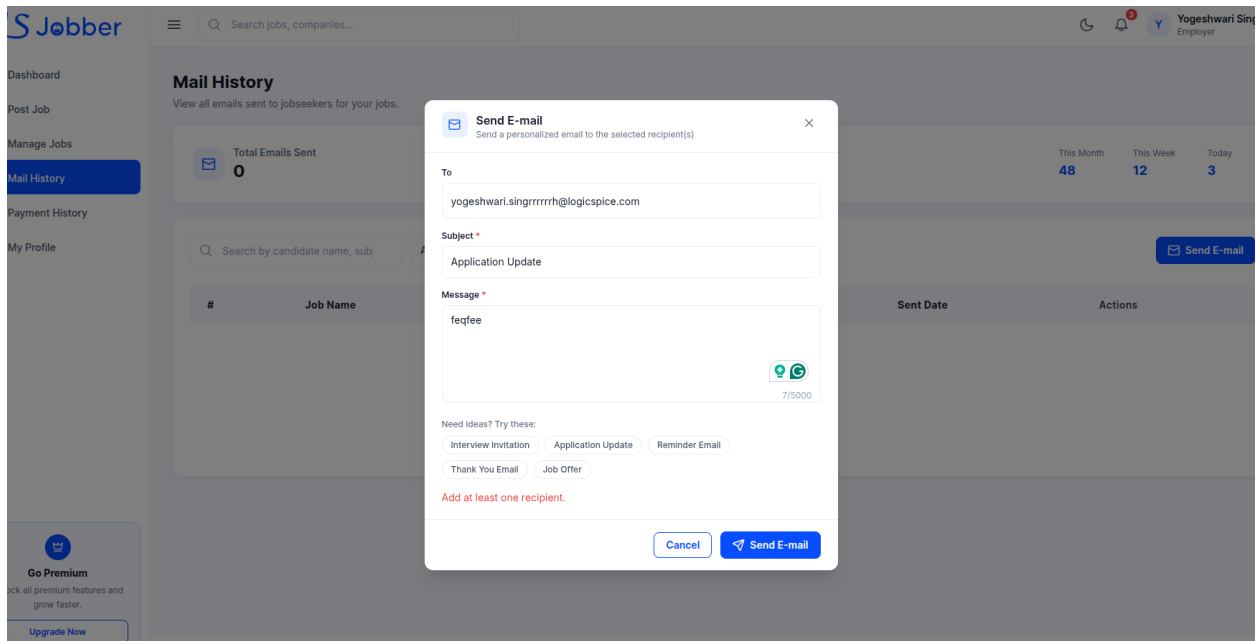
Expected Behavior:

- If a valid email is entered in the "To" field, the validation error should disappear.
- Clicking "Send E-mail" should proceed with the API call.
- The form should recognize the entered email as a valid recipient.

Please check:

1. Is the "To" field correctly bound to the recipients array/state?
2. Is the validation checking a different field than the visible input?
3. Is the recipient list being updated only after pressing Enter/comma or selecting from autocomplete?
4. Ensure manually typing a valid email is also treated as a recipient.
5. Verify that clicking "Send E-mail" triggers the API request in the Network tab.
6. If the API is not being called, identify where the validation is stopping the submission.

Please fix this issue so the validation and submission work correctly.



11. Please fix the Date Picker / Calendar UI globally.

Issues:

1. The calendar popup UI is not matching the overall design system.
2. There is an unwanted border around the calendar popup.
3. The calendar icon has no proper background, padding, or alignment. It should match the design of all other input suffix icons.
4. Spacing, border-radius, shadow, typography, and colors should follow the common UI used across the project.
5. The popup should have a clean shadow and no extra borders.
6. Ensure proper z-index so it always appears above other elements.

Refactoring Required:

- Create one ****Common DatePicker component**** (or common wrapper) and use it everywhere in the project.
- Remove all duplicate/custom DatePicker implementations from individual pages.
- Every screen should use this single reusable DatePicker component.
- Keep the existing functionality exactly the same. Only improve the UI and centralize the implementation.

Common Component Requirements:

- Same border radius as other inputs.
- Same input height and padding.
- Calendar icon with proper background, spacing, and hover state.
- Consistent popup width and styling.
- Responsive on desktop, tablet, and mobile.
- Dark theme compatible.

- Do not change any business logic or validations.

After creating the common component, replace all repeated calendar implementations throughout the project with this reusable component so future changes only need to be made in one place.

The screenshot shows the 'Post Job' form in the LS Jobber application. The form includes fields for 'Select Work Type', 'Designation', 'Experience (in Years)', 'Annual Salary', and 'Number of Openings'. A calendar modal is open for the 'Add Expiration Date' field, showing the month of July 2026. The calendar modal is highlighted with a red border. The form also includes a 'Go Premium' button and a 'My Profile' link. The footer shows the copyright notice '© 2026 LS JobBoard. All rights reserved.' and the text 'Developed by LogicSpice'.

LS Jobber

Search jobs, companies...

Yogeshwari Singh
Employer

Dashboard

Post Job

Manage Jobs

Mail History

Payment History

My Profile

Select Work Type

Designation *

Choose Designation

Experience (in Years) *

Choose Experience

Annual Salary *

Select Salary

Number of Openings *

1

Add Expiration Date *

mm/dd/yyyy

July 2026

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

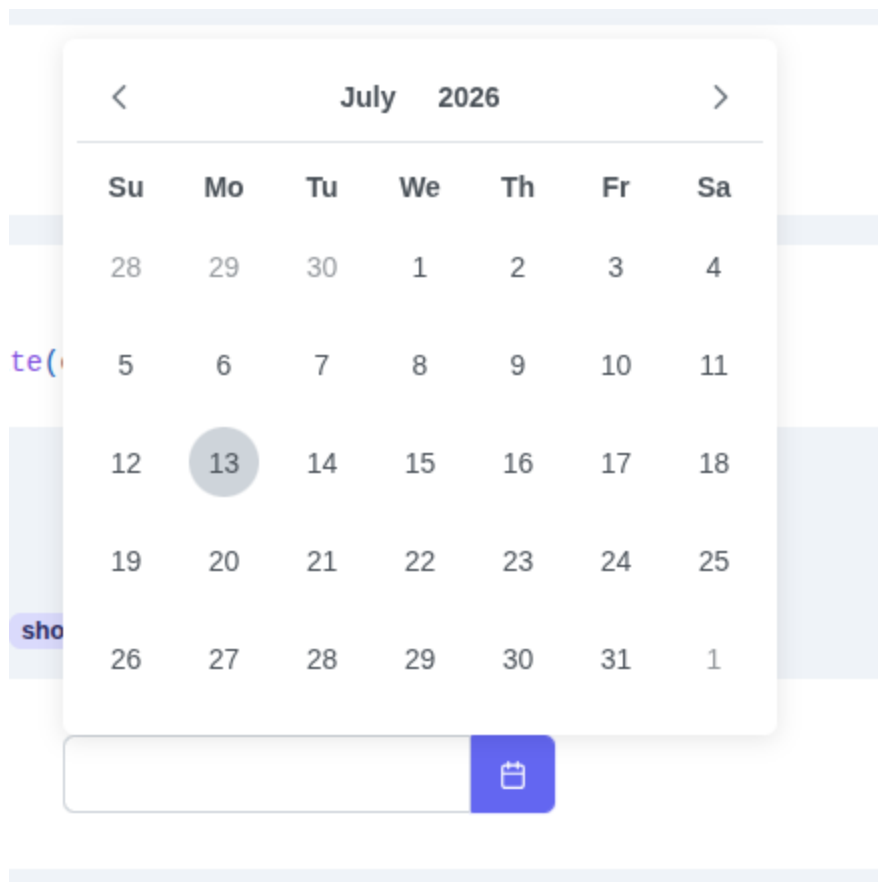
About the Role

Normal Sans Serif B I U A

© 2026 LS JobBoard. All rights reserved.

Developed by LogicSpice

EXPECTED:



12. Please check and fix the Post Job page dropdowns.

Issue:

The **Category** and **Sub Category** dropdowns are empty and no options are being displayed.

Expected Behavior:

1. **Category**

- Category options should be fetched dynamically from the Admin side Category API.
- Whatever categories the Admin creates should automatically appear here.
- The dropdown should not show "No available options" if data exists.

2. **Sub Category**

- After selecting a Category, the Sub Category dropdown should load dynamically based on the selected Category.
- Fetch the data from the appropriate API instead of using static or empty data.

3. Please verify:

- Is the API being called?

- Is the response coming correctly?
- Is the response being mapped to the dropdown options?
- Are there any frontend filtering or state issues preventing the options from rendering?

4. If the Admin has not created any Category/Sub Category yet, show a proper empty state message instead of a broken dropdown.

5. Also check all other dropdowns on the Post Job page.

- Any dropdown whose data should come from the Admin panel must be connected dynamically to its respective API.
- Any dropdown that already has its own API should fetch and display data correctly.

Please identify which dropdowns depend on Admin-managed data and which are populated directly via APIs, then fix all integrations accordingly.

Do not use hardcoded or dummy data. Everything should be loaded dynamically from the backend.

13. Please update the global "No Data Found" UI across the entire project.

Current Issue:

Wherever a table has no records, an old/default empty-state icon is being shown.

Required Changes:

1. I have uploaded a new ****No Data**** illustration. Use this image instead of the current default icon.
2. Replace the old empty-state icon with the uploaded illustration everywhere.
3. This should be a ****global/common component****, not implemented separately on each page.
4. Every table in the project should use the same common empty-state component.

This includes (but is not limited to):

- Manage Jobs
- Mail History
- Payment History
- Applications
- Candidates
- Notifications
- Staff Management
- Roles & Permissions
- Newsletter
- Blogs
- CMS tables
- Website Settings tables
- Search History

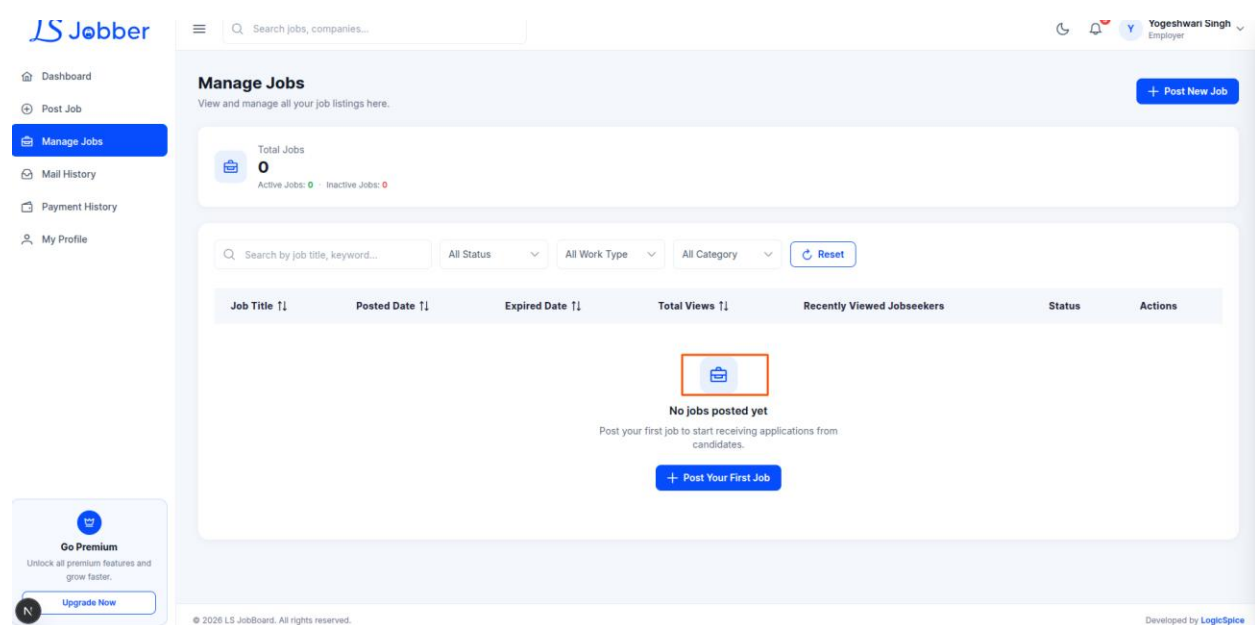
- Inquiries
- Payment History
- All Admin tables
- All Employer tables
- All Job Seeker tables
- Any other page where a table/list has no records

Design Requirements:

- Show the uploaded ****No Data**** illustration.
- Keep the title and description relevant to each module (or use the existing text if it's already appropriate).
- Keep any existing CTA button (e.g. "Post Your First Job", "Add Staff", etc.) exactly as it is.
- Only replace the empty-state icon with the new illustration.
- The UI should remain responsive and work in both Light and Dark themes.

Refactoring:

- Create a reusable ****Common Empty State**** component.
- Remove all duplicated "No Data" implementations from the project.
- Every table/list with no data should use this single common component so future updates only need to be made in one place.



13. One more important issue needs to be investigated and fixed.

Current Issue:

I am logged in as the employer account ****yogeshwari.singh****. This employer has not posted any jobs yet, but jobs are still appearing on the public website.

Please investigate this completely.

Things to verify:

1. Check why jobs are appearing on the public website even though this employer has not posted any jobs.
2. Verify whether the website is showing:
 - Dummy/static data
 - Mock data
 - Jobs from another employer
 - Incorrect API response
 - Incorrect filtering based on employer_id
3. Ensure that the public website only displays jobs that actually exist in the database and meet the required conditions (published/active, etc.).
4. When I log in with my employer account and successfully post a job, only then should my job appear on the public website according to the backend logic.
5. If the jobs currently shown are dummy/mock data, remove them and replace them with dynamic API data.

Also, after completing all the issues I have reported today:

- Pull the latest Employer branch.
- Merge the latest Job Seeker changes if required.
- Resolve all reported issues.
- Test the complete Employer flow from start to finish.
- Verify that there are no console errors or failed API requests.
- Test the complete job posting flow, job listing, public website display, mail flow, dropdowns, dashboard, and all related functionality.

Finally, provide me with a detailed report containing:

- All issues that were fixed.
- Any issues that were caused by missing backend APIs.
- Any remaining blockers (if any).
- APIs integrated or updated.
- Components that were made common/reusable.
- Any assumptions or temporary changes made.
- Confirmation that everything has been tested locally.

Please share the final report only after verifying all functionality end-to-end.

14. Please improve the UX of the Manage Jobs page when there is no data.

Current Issue:

When there are no jobs available, all the search and filter controls are still enabled, even though they have nothing to filter.

Required Behavior:

1. If the jobs table is empty (0 records):

- Disable the Search input.
- Disable all filter dropdowns (Status, Work Type, Category, etc.).
- Disable the Reset button.
- Show a `not-allowed` cursor (`cursor: not-allowed`) on all disabled controls.
- Reduce the opacity slightly to indicate they are disabled.
- Prevent any interaction with these controls.

2. Once at least one job exists:

- Automatically enable all search and filter controls.
- Restore their normal appearance and functionality.

3. This behavior should be implemented dynamically based on the table data count, not with hardcoded conditions.

4. Apply the same UX across the project:

- Wherever a table has no records, disable its related search, filters, bulk actions, and reset controls.
- Automatically enable them when data becomes available.

This will provide a much better user experience and avoid allowing users to interact with filters that have no data to operate on.

Manage Jobs

View and manage all your job listings here.

+ Post New Job

Total Jobs
0
Active Jobs: 0 Inactive Jobs: 0

Search by job title, keyword... Inactive Full-time Design Reset

Job Title	Posted Date	Expired Date	Total Views	Recently Viewed Jobseekers	Status	Actions
-----------	-------------	--------------	-------------	----------------------------	--------	---------



No jobs posted yet

Post your first job to start receiving applications from candidates.

+ Post Your First Job



Go Premium

Unlock all premium features and grow faster.

Upgrade Now