

Frontend Demo Presentation (By John)

Introduction

Good morning Sir and everyone,

I'm John, and I was responsible for the frontend development in our project team. Today, I'll walk you through the main features I built using React.js and Tailwind CSS to ensure a secure, responsive, and user-friendly experience for everyone.

Slide 1: Multilingual Landing Page

- Clean and modern UI
- Built-in language switcher (English / Japanese)
- Powered by i18next
- Seamless switching without page reload

Explanation While Showing:

This is the landing page. You'll notice the clean layout and how easy it is to navigate. Up here, we have the language switcher. Users can instantly switch between English and Japanese using i18next, and the page updates without reloading. This helps us serve both local and international users more effectively.

Slide 2: Real-time Course Catalog

- Displays latest updates instantly
- Fully responsive (desktop and mobile)
- Cards or list format for clear course display

Explanation While Showing:

This section shows the course catalog. You'll see it's fully responsive-whether on a phone or desktop, the layout stays clean and readable. This improves usability for all users. We built an interactive course browsing system with real-time filtering and search capabilities, allowing users to efficiently discover relevant educational offerings.

Frontend Demo Presentation (By John)

Slide 3: Guided Application Process

- Split into easy-to-follow steps
- Real-time validation
- Smooth transitions between steps
- Admin approval required for dashboard access

Explanation While Showing:

I created both an application form and a contact form. When users submit either form, the data is sent to the backend using Axios and is automatically displayed in the Admin Dashboard. This allows the admin to review submissions directly from the frontend-without needing to manually access the backend system.

We've broken the application form into multiple steps to make it easier to complete. Each field is validated in real time, ensuring users don't submit incomplete or incorrect information. After submission, the data is sent to the Admin Dashboard for approval.

Slide 4: Secure Access Control

- Login modal with role validation
- Routes users to: Student dashboard, Teacher dashboard, Admin dashboard
- Students must be approved by admin before access

Explanation While Showing:

Here's the login modal. Once a user logs in, we check their role and redirect them accordingly. If it's a student, they'll see the student dashboard-but only if they've been approved by an admin. This ensures only valid users can access secure areas of the site.

Slide 5: Student Registration Approval

- Students don't get access immediately
- Pending approval from the admin
- Only approved students can log in

Frontend Demo Presentation (By John)

- Ignored or rejected students remain blocked

Explanation While Showing:

This is a custom feature we added to improve security and control. When a student registers, they can't immediately access their dashboard. Instead, their request is sent to the admin. If the admin approves it, the student can log in. But if the admin ignores or rejects the request, the student stays locked out of the system. This prevents unauthorized access and gives admins more control.

Slide 6: Personalized Dashboards

- Student: Track progress, view assignments, access courses
- Teacher: Monitor student activity, manage content, post updates
- Admin: Approve users, manage data, oversee platform metrics

Explanation While Showing:

Each user sees a different dashboard. Students see their own learning data and materials. Teachers have tools to monitor and support student learning. Admins have full access-they can approve new students, manage roles, and view analytics. This keeps the system organized and secure.

Slide 7: Admin Messaging System

- Admin can send messages to each applicant
- Chat-style message box/modal
- Opens after receiving the application

Explanation While Showing:

After receiving an application, the admin can message the applicant directly from the dashboard. This allows for follow-up questions or sending updates without using email. It improves communication and speeds up decision-making. A simple chat modal opens for each applicant to support one-on-one communication.

Frontend Demo Presentation (By John)

Slide 8: Update or Delete Applications

- Edit application details if needed
- Delete any incorrect or unnecessary submission
- Keeps admin data clean and accurate

Explanation While Showing:

Admins now have full control over application data. If an applicant enters wrong information, the admin can edit it right from the dashboard. Or if the application is invalid, it can be deleted instantly. This ensures that only valid data remains in the system.

Slide 9: Contact Submissions + Status Control

- View all contact form messages
- Change status: Pending -> Approved -> Resolved
- Organized support tracking

Explanation While Showing:

We added a new section to manage contact form messages. Admins can see each message submitted through the contact page and change its status. For example, once a message is replied to, the admin can mark it as "Resolved". This keeps customer support organized and helps track pending tasks.

Slide 10: Conclusion

- Built with React.js and Tailwind CSS
- Multilingual, responsive interface
- Secure login with role-based access
- Smooth admin workflows and backend integration via Axios
- New features: Messaging, Edit/Delete, Contact Manager

Explanation While Showing:

Frontend Demo Presentation (By John)

This frontend is built using React and Tailwind CSS. We've made it multilingual, responsive, and securely connected to the backend with Axios. These features streamline user experience and simplify admin workflows. With the new features-like messaging applicants, editing submissions, and tracking contact messages-we're making the platform even more powerful and user-friendly.

Thank you for your time - this concludes my demo highlights for the frontend. Now, I'll hand it over to ****Shashini**** to explain the other parts of our project. Thank you!