

Our front-end is implemented mainly by these techniques:

Node.js, which is a JavaScript runtime environment; **Vue of version two**, a popular JavaScript framework; **Element UI**, a Vue-based component library; **Vue Router**, the official routing library for Vue.js; **Axios**, an HTTP client used for asynchronous data fetching and communication with backend servers in frontend applications. Then based on these techniques, we have implemented four webpages.

Firstly, login page.

It has the title, icons, and a login form highlighted on the background picture.

There are three buttons: Log in, Reset, and Register. **Log in** checks if the username and password are filled. When you click Log in, it checks the form, sends a request to the server, and saves your information if it's correct. **Reset** button clears the form, and "**Register**" directs users to the next page.

Register page:

It's just a common style. Users input their name and password.

Axios is used for sending requests to the backend API for registration. **Custom validation functions** in the code ensure password requirements and matching confirmation.

With a user account, we can come into the panel page after login.

It splits into three sections, allows uploading files, and includes a Log Out button for easy exit.

Users can switch between viewing new and recent uploads, which will be discussed later.

CSS Grid ensures a visually appealing. Session storage maintains user information across visits for a personalized experience.

A dialog box "IUPLOAD PLATFORM" will appear when clicking the button "Upload files".

The upload component allows users to either drag and drop (multiple) files or click to upload them, which will appear below for a double check.

The dialog box includes buttons for canceling the upload operation and submitting the uploaded files.

After successful uploading, files with its information show together with hint messages.

The session in the middle is divided into two categories, allowing users to switch between viewing New Upload or Recent files by clicking on buttons.

New Upload includes only files uploaded during the current session; **Recent files** consist of all files uploaded by the user recently.

Each file's name, upload time, and file size are displayed on the page.

As for the **implementation**, the upload page receives user data as a **prop**, logs it on mount, and handles successful file uploads with formatted information. The **handleUpload** method manages the upload process by clearing old uploads, submitting files, and ensuring a smooth user experience with dialog visibility control.