Education Web App Frontend (Using React)

1. User Authentication and Role-Based Views:

Integrate React components for user registration and login pages.

Implement user authentication features using sessions or tokens to ensure secure access.

Create different views for Educators and Students after successful authentication.

2. Educator Views:

2.1. Educator Dashboard:

Display a personalized dashboard for Educators upon login.

Include an overview of their uploaded resources, recent activities, etc.

2.2. Educator Profile Page:

Create a dedicated page for Educators to view and edit their profile details.

Implement a form allowing Educators to update their information.

2.3. Resource Upload Page:

Build a page where Educators can upload various types of educational resources.

Implement a file upload button to facilitate resource uploads.

Include fields for resource details like subject, branch, semester, description, and tags.

2.4. Resource Management Page:

Develop a page where Educators can view, edit, and delete their uploaded resources.

Implement functionality to organize and categorize resources.

3. Student Views:

3.1. Student Dashboard:

Create a personalized dashboard for Students upon login.

Provide an overview of available resources, recent activities, etc.

3.2. Student Profile Page:

Build a page for Students to view and edit their profile details.

Implement a form allowing Students to update their information.

3.3. Resource Download Page:

Develop a page where Students can browse and download educational resources.

Implement filters and search options for resources by subject, branch, and semester.

Enable students to view resource details and download associated files.

Integration with Backend:

Connect the React frontend with the previously constructed backend.

Ensure smooth communication between the frontend and backend for user authentication, resource uploads, downloads, etc.

Concepts to implement:

React core, Routing, State management, React Hooks, props, file management with react, connection with backend, etc

Additional Guidance:

- Make effective use of React components and state management for a modular and maintainable codebase.
- Implement responsive design principles to ensure a seamless user experience across different devices.
- Consider the user interface design, keeping in mind the user journey and ease of use.
- Remember, this assignment will showcase your skills in frontend development, so pay attention to details and aim for a polished and user-friendly interface.
- Feel free to leverage any additional technologies or libraries that you find suitable for this task. If you have any questions or need clarification on specific requirements, don't hesitate to reach out.

We are excited to see your progress on the frontend implementation. Good luck, and happy coding!