Creating our own NRCS

Aim: Understanding the creation of a NRCS (Newsroom Computer System)

Steps:

Step 1: Understand the Requirements

1. **Define Objectives:**
    - Identify the core features your NRCS needs, such as user management, news story creation, editing, scheduling, and media management.
2. **Assess Technical Skills:**
    - Ensure your team has the necessary skills in web development, database management, and possibly some familiarity with broadcasting protocols like MOS.

Step 2: Set Up Your Development Environment

1. **Choose a Technology Stack:**
    - Select the appropriate technologies for your project. For example:
        - **Backend:** PHP, Python, or Node.js
        - **Frontend:** HTML, CSS, JavaScript (React, Vue, or Angular)
        - **Database:** MySQL, PostgreSQL, or MongoDB
2. **Install Development Tools:**
    - Install a local server environment like XAMPP, MAMP, or Docker.
    - Set up version control with Git.

Step 3: Explore EBU's Awesome Broadcasting Resources

1. **Review the Repository:**
    - Go through the Awesome Broadcasting repository to identify useful libraries, frameworks, and tools.
2. **Select Relevant Tools:**
    - Identify tools and libraries that align with your NRCS requirements. For example:
        - **Newsroom Systems:** Tools for managing news stories, rundowns, and media.
        - **Media Management:** Solutions for handling video, audio, and graphics.
        - **Integration Tools:** Libraries for integrating with broadcast equipment and protocols like MOS.

Step 4: Design Your NRCS

1. **System Architecture:**
   - Design the architecture of your NRCS. This includes defining how different components (frontend, backend, database) will interact.
2. **User Interface Design:**
   - Create wireframes and mockups of the user interface. Consider user roles (e.g., journalists, editors, producers) and their workflows.

Step 5: Develop Core Features

1. **User Management:**
   - Implement user registration, login, and role-based access control.
2. **Story Creation and Editing:**
   - Develop features for creating, editing, and saving news stories.
3. **Rundown Management:**
   - Implement functionality for creating and managing rundowns (news bulletins).
4. **Media Management:**
   - Develop a system for uploading, organizing, and retrieving media assets.

Step 6: Integrate Broadcasting Protocols

1. **MOS Protocol:**
   - Integrate the MOS protocol to enable communication between your NRCS and broadcast devices. Use libraries and examples from the EBU repository.
2. **Other Protocols:**
   - If needed, integrate other broadcasting protocols relevant to your system.

Step 7: Testing and Quality Assurance

1. **Unit Testing:**
   - Write unit tests for your application to ensure each component works as expected.
2. **Integration Testing:**
   - Test the integration between different components of your NRCS.
3. **User Acceptance Testing:**
   - Conduct testing with actual users to gather feedback and identify any issues.

Step 8: Deployment

1. **Set Up a Production Environment:**
   - ○ Prepare a production server environment with the necessary configurations and security measures.
2. **Deploy the Application:**
   - ○ Use tools like Docker, Kubernetes, or a cloud platform (AWS, Azure, etc.) to deploy your application.
3. **Monitor and Maintain:**
   - ○ Set up monitoring tools to track the performance and health of your NRCS. Regularly update and maintain the system to ensure smooth operation.

Step 9: Documentation and Training

1. **Document the System:**
   - ○ Create comprehensive documentation for users and developers. Include installation guides, API documentation, and user manuals.
2. **Training:**
   - ○ Provide training sessions for your team and end-users to ensure they understand how to use the NRCS effectively.

Step 10: Continuous Improvement

1. **Gather Feedback:**
   - ○ Continuously collect feedback from users to identify areas for improvement.
2. **Iterate and Improve:**
   - ○ Regularly update and improve your NRCS based on feedback and changing requirements.