Assessment Task

Overview

Build a multi-step user registration system with a React based frontend. This assignment will help us evaluate your technical skills, problem-solving approach, and code quality.

Project Description

Create a **User Registration** application that collects user information across multiple steps and submits the complete data at the end of the process.

Frontend Requirements

1. **Multi-Step Form** with 3 steps:
    - **Step 1: Personal Information**
        - First Name (required)
        - Last Name (required)
        - Email (required, must be valid email format)
        - Phone Number (optional, must be valid format if provided)
    - **Step 2: Address Details**
        - Street Address (required)
        - City (required)
        - State/Province (required, dropdown selection)
        - Country (required, dropdown selection)
    - **Step 3: Account Setup**
        - Username (required, minimum 6 characters, check availability)
        - Password (required, minimum 8 characters, must include uppercase, lowercase, number, and special character)
        - Confirm Password (required, must match password)
        - Terms and Conditions checkbox (required)
        - Newsletter subscription (optional checkbox)
2. **Navigation Features**
    - Progress indicator showing current step and can be used to jump to the step
    - Previous/Next buttons for navigation between steps

    - Ability to go back and edit previous steps
    - Data persistence when navigating between steps
3. **Validation Requirements**
    - Real-time field validation with appropriate error messages
    - Prevent progression to next step if current step has validation errors
    - Validation for all fields
4. **Final Submission**
    - Review screen showing all entered information before final submission
    - Submit button that sends all collected data to the API endpoint
    - Success/Error handling with appropriate user feedback
    - Reset form after successful submission
5. **UI/UX Requirements**
    - Loading states during async operations
    - Clean and intuitive user interface
6. **Bonus**
    - Clean and intuitive user interface
    - Validate Step 2 : Make sure user's email matches the email domain (`eg. if user selects country as UK their email should include .uk`)

Backend Requirements

Build a backend service to support the multi-step user registration process described above. Your backend should meet the following requirements:

Functional Requirements

1. **Monorepo Structure**
    - Place both the backend (Go) and frontend (React) in a single repository.
    - The backend should serve the frontend's static assets, or you may decouple them if you prefer.
2. **API Endpoints**
    - `POST /api/register`: Accepts the complete registration payload after all steps are completed.
        - Checks that the email is available before storing user data.
    - Design an endpoint to check username availability
3. **Form Response Handling**
    - Accept a single JSON payload representing all registration steps.

   - Parse, validate, and report errors in a structured format.

4. **Validation**
   - Field-level validation (e.g., email format, password strength).
   - Cross-field validation (e.g., password confirmation, country/email domain match).
   - Business logic validation (e.g., email uniqueness, username uniqueness).
   - Implement validation as a middleware chain, where each validator is a middleware function. It should be easy to add or remove validators without breaking the app.
5. **Middleware Chain Support**
   - Middlewares should be registered in a way that allows easy addition/removal without breaking the chain.
   - Include middleware for logging, extra validation.
6. **Database Storage**
   - Persist user registration data to a database of your choice.
   - Store all fields from the registration process, including audit fields (e.g., created_at, updated_at).
7. **Error Handling & Feedback**
   - Return consistent JSON error responses with error codes and messages.
   - Return field-specific validation errors for frontend display.
   - On successful registration, return a success message and (optionally) a user ID.
8. **Bonus Features**
   - Configurable database backend via environment variables or config file.
   - (Optional) Health and readiness endpoints for deployment checks.
   - Bonus: Make it easy to switch to a different database system (limited to SQL) if needed.

TASK

Technical Requirements

Tech Stack

- **Frontend Framework**: React
- **State Management**: Your choice (eg. Context API, Redux)
- **Styling**: Your choice (eg CSS , SASS)
- **Build Tool**: Vite or Create React App

Deliverables

1. **Source Code**
   - Create a new GitHub repository

    - Share the repository link upon completion
    - Include a comprehensive README.md with:
        - Project setup instructions
        - How to run the application
        - How to run tests
        - Brief architectural overview
        - Any assumptions made
2. **Documentation**
    - README with clear setup instructions

Time Expectation

This assignment should take approximately 4-6 hours to complete. Please submit your solution within 5 days of receiving this assignment.

Submission Instructions

1. Complete the implementation in your own GitHub repository
2. Ensure all code is committed and pushed
3. Share the repository link via email
4. Include any additional notes or explanations in the README

Notes

- Focus on demonstrating your best practices and coding standards
- You may use any additional libraries you feel are necessary
- If you make any significant assumptions, document them in the README

Questions?

If you have any questions or need clarification on any requirements, please don't hesitate to reach out.

Good luck!