FORSTU: Full Stack Web Developer Coding Assignment

Time Duration: 3 Days

Task 1: Excel Data Importer

Background Context: As part of our scholarship management system, we need a Node.js script that can efficiently handle the import of student data from an Excel file provided by the college.

Use Case:

Imagine the college provides an Excel sheet with details of enrolled students. The script should seamlessly read this file and generate student profiles, ensuring all necessary information is captured for further scholarship processing.

Task Description:

1.Read Excel File:

- Develop a script to read the provided Excel file.
- Extract relevant information such as student names, emails, and enrollment dates.

2.Create Student Profiles into Database:

- Generate student profiles in database using the extracted data.
- Include default fields for all students in the profiles for. E.g set Default State to Maharashtra in Database for all students imported through excel.

Develop a Node.js script that can import student data from an Excel file provided by the college. The script should:

- Read the Excel file and extract relevant information (e.g., name, email, enrollment date).
- Create student profiles with default fields for all students.

Task 2: Dynamic Form Builder

Background Context:

In our scholarship management system, we require a dynamic form that adapts to each student's null values This form aims to gather additional information & pending essential for accurate scholarship processing.

Use Case:

Consider a scenario where a student, after being profiled from the Excel data, receives an email notification to fill out a dynamic form tailored to their specific scholarship eligibility criteria. The form adjusts dynamically based on the student's existing data, providing a smooth and guided process.

Task Description:

1. **Dynamic Field Display:**

   Develop a React component that dynamically displays form fields based on the s student's pending data.

2. **Email Notifications:**

   Implement functionality to automatically send email notifications to students, prompting them to complete the dynamic form.

3. **Form Validation & Submission:**

   Incorporate form validation to ensure students provide accurate and valid information and also provide ability to submit the form.

4. **Progress Tracker:**

   Include a progress tracker within the form, guiding students through the step-by-step process of filling out the required information.

Task 3:Scholarship Assignment Algorithm

Background Context:

In our scholarship management system, we need a robust Node.js algorithm that automates the process of determining scholarship eligibility for students based on a set of criteria. This algorithm will play a crucial role in efficiently assigning scholarships.

Use Case:

Imagine a scenario where a large pool of student data needs to be processed to identify eligible candidates for various scholarships. The algorithm should efficiently sift through this data, applying predefined criteria to automatically assign scholarships to deserving students.
Assume the criteria and define the scholarship as per it.

Task Description:

Eligibility Criteria:

Define and understand the eligibility criteria for scholarships, considering factors such as academic performance, financial need, and extracurricular activities.

Algorithm Implementation:

Develop a Node.js algorithm that evaluates student data against the predefined eligibility criteria.

Optimization:

Design the algorithm for optimal performance, considering factors such as speed and efficiency in searching and assigning scholarships.

Automation:

Automate the scholarship assignment process using the implemented algorithm.

Task 4: Interactive Dashboard: Admin Panel (React)

Background Context:

As part of our scholarship management system, we need to augment the React application with an interactive dashboard tailored for administrators. This dashboard will provide a comprehensive overview of scholarship assignments and empower administrators with tools for manual adjustments.

Use Case:
Consider an administrator accessing the dashboard to review scholarship assignments. The summary provides key insights, and the ability to manually adjust assignments allows for customization based on specific considerations. Visualizations enhance the understanding of scholarship distribution patterns.

Task Description:

Summary Display:

Create a React component that displays a concise summary of scholarship assignments, offering a quick snapshot for administrators.

Manual Adjustment Feature:

Implement functionality to allow administrators to review and manually adjust scholarship assignments within the dashboard.

User Authentication:

Implement user authentication specifically tailored for administrators, ensuring secure access to the dashboard features.

Submission:

1. You can submit the code on GitHub or alternatively also host it on any temporary domain also.