## Page:1

# Full-Stack Website Development with Basic Authentication and Role-Based Authorization

**Instructions**

**In this particular problem statement, you are required to develop a fully-fledged full-stack website. You can use any frontend technology, whether it be React or plain HTML, CSS, and JavaScript. The goal is to learn frontend and backend development simultaneously, so feel free to be as creative as you want.**

**Navigation Bar**

* The navigation bar should contain the following page links:
  + Home Page
  + About Page
  + Sign Up Page
  + Login Page
* The navbar should be visible across all pages.

**Home Page**

* This page should feature a welcome note, providing a chance to practice your writing skills.
* You can be as creative as you want, using images or GIFs, for example.

**About Page**

* This page should include all the details about your project, such as the pages, features, and packages used.
* It can also showcase screenshots of your project.

**Sign Up Page**

* The registration form should include the following fields:
  + Username (Input Field)
  + Email (Input Field)
  + Date of Birth (Input Field)
  + Role (Dropdown Menu: Admin, Explorer)
  + Location (Input Field)
  + Password (Input Field, Type: Password)
  + Confirm Password (Input Field, Type: Password)

## Page:2

* The user should be registered only if the password and confirm password fields contain the same string.
* These user details should be stored in MongoDB .

**Login Page**

* The login form should include the following fields:
  + Username
  + Password
* Authentication during login should be handled using a middleware.
* After successful authentication, the user should be redirected to the home page.
* The username should be visible on the navigation bar, regardless of which page the user navigates to.
* A logout button should also appear on the navbar; clicking it should log the user out and remove the username from the navbar.

**Routes**

* POST: To register user details during sign-up.
* GET: To retrieve all user details.
* GET: To retrieve details of a particular user.
* PATCH/PUT: To update user details (Only Admin can do this).
* DELETE: To delete a particular user’s details (Only Admin can do this).

**Middlewares**

* **Authenticator**: Authenticates the user during login and allows them to log in.
* **Validator**: Validates the role of the user when updating or deleting user details.
* **UserLogger**: Logs the username and role of the user to a log.txt file upon successful login.