Building a Full-Stack User Management System Similar to Instagram

Assignment

Objective:
The objective of this assignment is to create a full-stack user info and login system that resembles the functionality of Instagram. The system will be built using HTML, CSS, and JavaScript for the front-end, while the back-end will be developed using Express.js and MongoDB.

Requirements:

Front-End Development:
a. Design a visually appealing user interface using HTML and CSS.
b. Create an intuitive user registration form that captures essential details like username, email, and password & bio & Name
c. Develop a login form that authenticates users and verifies their credentials.

Back-End Development:
a. Use Express.js to handle HTTP requests and responses.
b. Set up a MongoDB database to store user information securely.
c. Implement user registration functionality by validating and saving user data to the database.
d. Develop a login mechanism that verifies user credentials against the stored data.
e: Generate JWT token and Store it into cache

Security:
a. Hash and salt user passwords before storing them in the database to enhance security.
b. Protect sensitive routes and ensure that only authenticated users can access them.

Documentation
a. Provide clear and concise documentation on how to set up and run the application.

Deliverables:
Complete source code of the front-end and back-end implementation.
A document containing instructions for setting up and running the system.
Note: You are encouraged to enhance the system beyond the specified requirements, add additional features, and improve the user experience.

Description & Api Endpoint

Backend:

"/signup" → create a signUp routes which takes {name,username,bio,email,password} as response and hash password and then store data in database

"signupDataValidate" → create a middleware which checks if user has provided required data or not based on that send response or move further

"/login" → create a login routes which takes {user,password} as response, generate jwt token and set to cookie and response with success message

"loginDataValidate" → create a middleware which checks if user has provided required data or not based on that send response or move further

"/" → create home route which will authenticate(hint: use authenticateUser middleware to achieve it) user and will send user data

"authenticateUser" → create authenticate user which verify token given by user through cache then then proceed based on the output
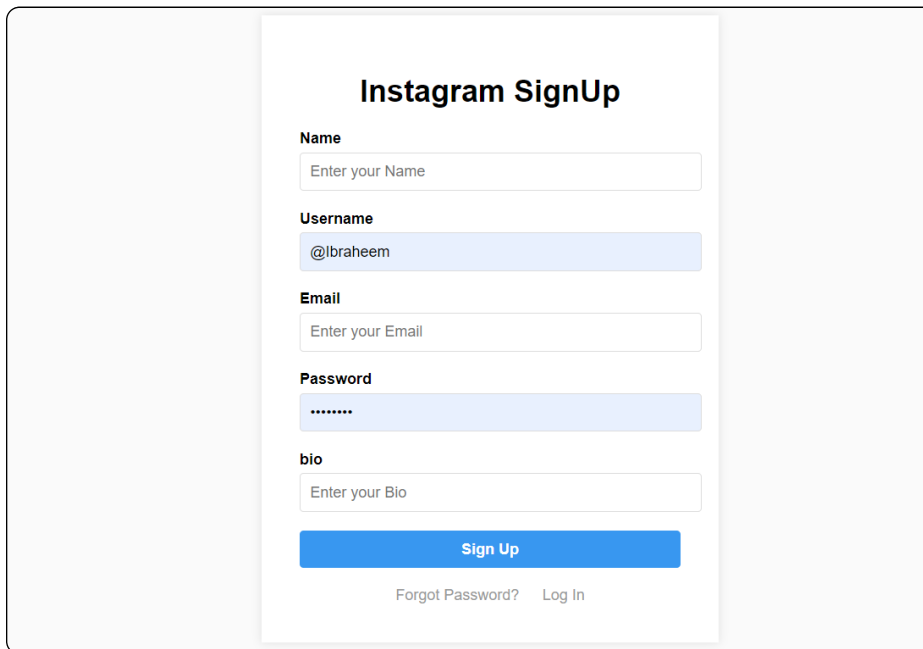
FrontEnd:

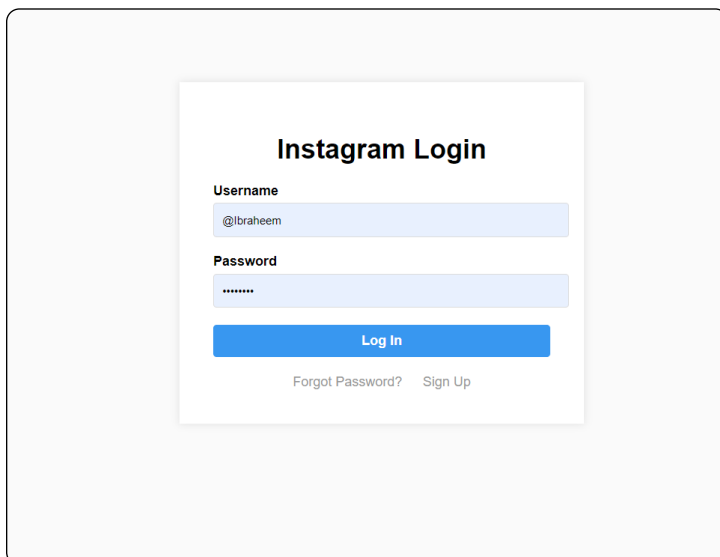**SignUp Page:** Create a sign up page after successful signup redirect to login page

Login Page: Build Login Page after successful Login Redirect to User Page

User Page: Create a user page which has demo image and rest of the data fetch from server by authenticating user

Images are given below for your reference

@Ibraheem

Content Creator | Sde - 1

ibraheem@gmaiil.com

Followers: 1000