**Client Requirement Document**

**Project Title:** CitiusTech Registration Page Development

**Client:** CitiusTech

**Prepared By:** Debanjan Das

**Date:** 01 Jan, 2025

**1. Project Overview**

CitiusTech requires a responsive and user-friendly registration page that allows users to input their details and register successfully. The registration form should capture the user's first name, last name, email, and phone number. Upon successful registration, users should receive a confirmation message and be redirected to the login page.

**2. Functional Requirements**

**2.1 User Interface (UI)**

* The page should have a clean and modern UI with a professional color scheme.
* The page background should be navy blue (#002A5C) with a white registration form.
* The form should be centered on the page and styled with proper padding and margins.
* The header should display "CitiusTech | Shaping Healthcare Possibilities."

**2.2 Registration Form**

* The form should include the following input fields:
  + **First Name** (Text Input, Required)
  + **Last Name** (Text Input, Required)
  + **Email** (Email Input, Required)
  + **Phone Number** (Telephone Input, Required)
* All fields must have appropriate placeholders and labels for better user understanding.
* The "Register" button should be visually distinct and change color on hover.

**2.3 Form Validation**

* The form should validate that all fields are filled before submission.
* The email input must follow a valid email format.
* The phone number input should accept only valid numerical values.

**2.4 Success Message and Redirection**

* Upon successful registration, a success message should be displayed: "Successfully registered! Redirecting to login..."
* After a 2-second delay, the user should be redirected to 'home.html'.

**2.5 Backend Integration**

* The form should send the user data to an API endpoint (http://localhost:3000/register) via a POST request.
* The data should be sent in JSON format with the following structure:
* {
* "first\_name": "UserFirstName",
* "last\_name": "UserLastName",
* "email": "user@example.com",
* "phone": "1234567890"

}

* If the server responds with an error, an alert message should be displayed: "Registration failed. Please try again."
* If there is a network issue, an alert should display: "Error connecting to the server."

**3. Non-Functional Requirements**

* The page should be fully responsive and accessible on desktop and mobile devices.
* The form must be lightweight and load quickly.
* The UI should follow modern web design principles for readability and usability.

**4. Technology Stack**

* **Frontend:** HTML, CSS, JavaScript
* **Backend API (for integration):** Node.js with Express (assumed)
* **Database (if applicable):** Not specified in this scope

**5. Deployment and Hosting**

* The frontend should be hosted on a web server or cloud platform.
* The backend API should be deployed on a Node.js server with database connectivity (if required).

**6. Deliverables**

* Fully functional and tested registration page.
* Frontend HTML, CSS, and JavaScript files.
* API integration for user registration.
* Deployment instructions and documentation.

**7. Timeline**

* **Design & Development:** [Specify Timeline]
* **Testing & Debugging:** [Specify Timeline]
* **Deployment:** [Specify Timeline]

**8. Acceptance Criteria**

* The registration form successfully captures user details and sends them to the backend.
* The success message is displayed after successful registration.
* Users are redirected to the login page upon successful form submission.
* The page is responsive and works on different devices.
* Error handling and validation work correctly.

**9. Additional Notes**

* Future enhancements could include password fields, reCAPTCHA, and multi-step registration processes.
* If database storage is required, discuss the data model with the backend development team.