COLLEGE CODE :9604

COLLEGE NAME : CSI Institute of Technology

DEPARTMENT : AI & DS

STUDENT NM-ID : aut960423243006

ROLL NO : 960423243006

DATE : 16-09-2025

TECHNOLOGY PROJECT NAME : User Registration and Validation

SUBMITTED BY,

NAME : Ashick Joshua C

MOBILE NO : 9487944311

1. Problem Statement

Many web applications allow users to sign up, but insecure or poorly validated registration forms can lead to incorrect data storage, duplicate accounts, or security risks such as SQL injection and weak password usage. The goal of this project is to create a secure, responsive **User Registration System** that validates user input on both client and server side, ensuring only correct and safe data is stored.

2. Users & Stakeholders

- **Primary Users**
  - New users creating accounts on a web application.
  - Administrators managing user data.
- **Stakeholders**
- End Users: People registering for access.
- Developers: Responsible for building and maintaining the registration system.
- Business Owners/Institutions: Require accurate and secure user data.

3. User Stories

- *As a new user, I want to register with my email and password so that I can access the application.*
- *As a user, I want real-time form validation so that I know if I enter invalid information.*
- *As an admin, I want unique user data stored so that duplicate accounts are avoided.*
- *As a user, I want password strength indicators so that I can create a secure password.*

4. MVP Features

- User can:
  - Fill in registration form fields (name, email, phone, password).
  - See real-time validation (e.g., valid email format, password strength.
  - Receive error messages for invalid or missing fields.
  - Submit form and have data stored in a database.
  - Get confirmation via email (optional enhancement).
- UI/UX:
  - Responsive design (mobile + desktop)
  - Clear field-level validation messages.
- Technical:
  - Front-end: HTML, CSS, JavaScript/React/Vue.

    - Back-end: Node.js/Python/PHP (with secure APIs).
    - Database: MySQL or MongoDB.
    - Password hashing (e.g., bcrypt).

5. Wireframes / API Endpoint List

Wireframes (Textual Representation)

- **Registration Page:**
  - Fields: Name, Email, Phone, Password, Confirm Password.
  - "Register" button and real-time validation messages.
- **Success Page:**
  - "Registration Successful" message and login link.

API Endpoint List (Future Integration)

- **POST /register** → Accepts and validates user data, stores it securely.
- **GET /users** → (Admin) Retrieve registered user list.

6. Acceptance Criteria

- The form validates all required fields before submission.
- Email addresses must follow standard format and be unique.
- Passwords must meet strength criteria (min length, mix of characters).
- Data is securely stored (hashed passwords).
- Responsive design works smoothly across devices.
- Displays success/failure message after registration.