Infosys Springboard Virtual Internship
“Developing chatbot on global wellness”

Milestone 1 - User Authentication and Profile Management

Name: Puneeth Reddy Tapasi

Date: 28-08-2025

User Authentication and Profile Management

1. Introduction

The task was to create a **Wellness Chatbot** using **Streamlit** without needing CSS, HTML, or external frontend code.Streamlit is a Python library that helps developers quickly build interactive web applications. This project provides a **secure user registration and login system**, stores profile details in an **SQLite database**, and allows users to **manage their profile** by updating personal information and selecting their **preferred language (English or Hindi)**. The application demonstrates how Streamlit can be extended beyond simple dashboards to handle **authentication, database storage, and multilingual support**.

---

2. Objectives

- Implement a secure **user registration system** with fields like Name, Age, Gender, Email, and Password.
- Provide a **login system** with authentication using hashed passwords.
- Store user information in an **SQLite database (wellness.db)**.
- Allow users to **update profile details** (name, age, gender, language).
- Enable users to **change password** after login.
- Provide **multilingual support** (English and Hindi) for UI elements.
- Design a clean and modern **UI/UX** with Streamlit card-style layout.

---

3. Code Implementation

```
import streamlit as st

import sqlite3

from passlib.hash import sha256_crypt

# Database setup

conn = sqlite3.connect("database/wellness.db")

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS users (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    email TEXT UNIQUE
```

```
    password TEXT,
    full_name TEXT,
    age INTEGER,
    gender TEXT,
    language TEXT
)
""")
conn.commit()
# Registration
def register_user(email, password, full_name, age, gender, language):
    hashed = sha256_crypt.hash(password)
    try:
        cursor.execute("INSERT INTO users (email, password, full_name, age, gender, language)
VALUES (?, ?, ?, ?, ?, ?)",
                       (email, hashed, full_name, age, gender, language))
        conn.commit()
        return True
    except sqlite3.IntegrityError:
        return False
# Login
def login_user(email, password):
    cursor.execute("SELECT id, password FROM users WHERE email=?", (email,))
    user = cursor.fetchone()
    if user and sha256_crypt.verify(password, user[1]):
        return user[0]
    return None
# Profile Management
def get_user(user_id):
```

```
    cursor.execute("SELECT * FROM users WHERE id=?", (user_id,))
    return cursor.fetchone()


def update_user(user_id, full_name, age, gender, language):
    cursor.execute("UPDATE users SET full_name=?, age=?, gender=?, language=? WHERE
id=?",
                   (full_name, age, gender, language, user_id))
    conn.commit()
```

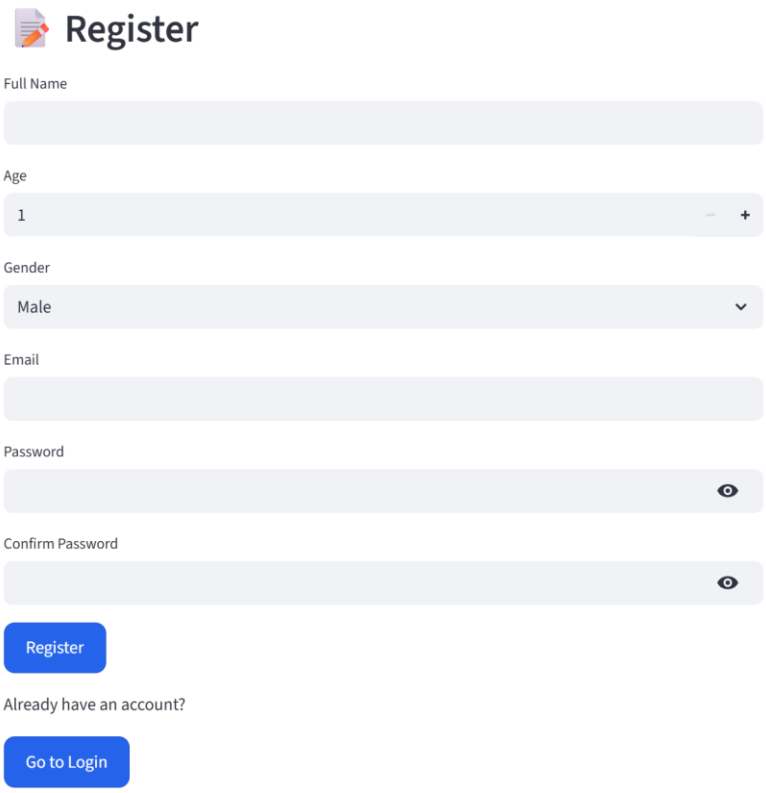
---

4. Explanation of Code

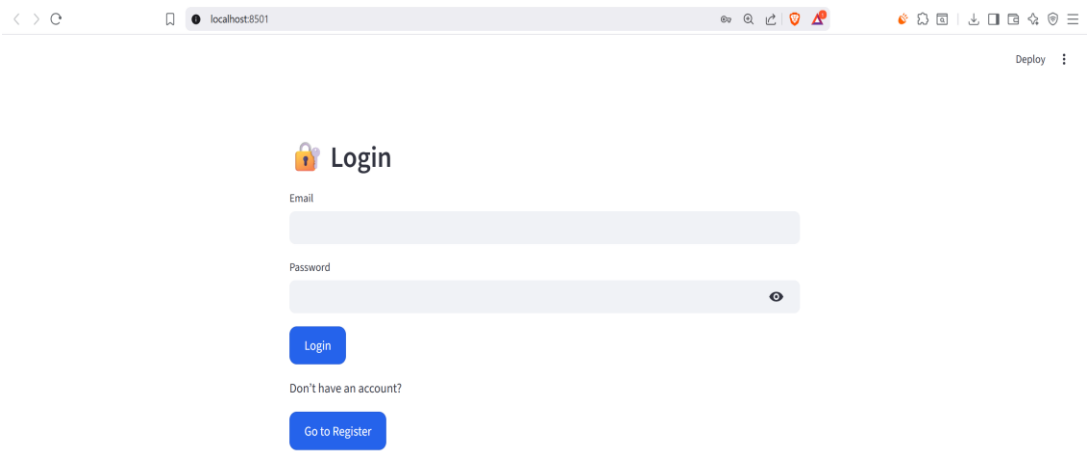
- **import streamlit as st →** Imports the Streamlit library to build the web interface.
- **import sqlite3 →** Connects to the SQLite database (wellness.db) for storing user data.
- **from passlib.hash import sha256_crypt →** Provides secure password hashing for registration and login.
- **Database Initialization**
    - A users table is created with fields: id, email, password, full_name, age, gender, language.
    - Ensures persistence of user accounts.
- **Registration Function (register_user)**
    - Accepts **Name, Age, Gender, Email, Password, Language.**
    - Hashes the password using sha256_crypt.
    - Stores the user details in the database.
    - Returns False if the email already exists.
- **Login Function (login_user)**
    - Verifies user credentials with sha256_crypt.verify.
    - Returns the user's ID if login is successful.
- **Profile Management (get_user, update_user)**
    - **get_user →** Retrieves stored profile details of the logged-in user.
    - **update_user →** Updates Full Name, Age, Gender, Language in the database.
- **Streamlit Pages (Frontend)**
    - **Register Page →** Form with Name, Age, Gender, Email, Password, Confirm Password, Language.
    - **Login Page →** Email and password input fields with authentication.
    - **Profile Management Page →** Displays user details, allows profile updates, and password change.
- **Language Preference**
    - Users can select English or Hindi.
    - The interface text changes dynamically based on the saved preference.
- **Session State**
    - Maintains whether a user is logged in or logged out.
    - Allows smooth navigation between Register → Login → Profile.

5. Output Screenshots

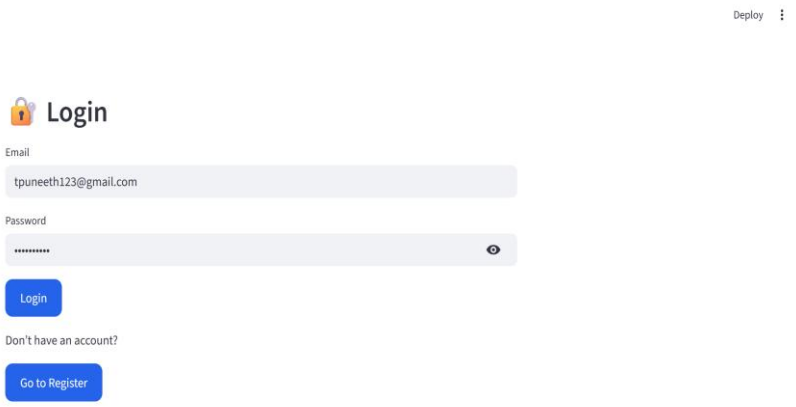
Step-1: Register Your details



Step-2: After completion of your Register, Login Page will be opened

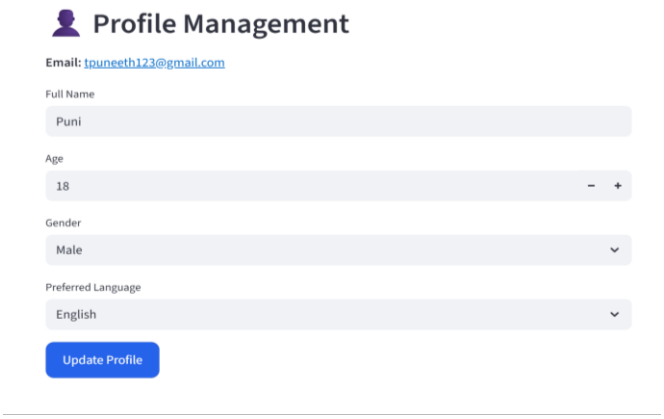


Step-3: Enter Login Credentials



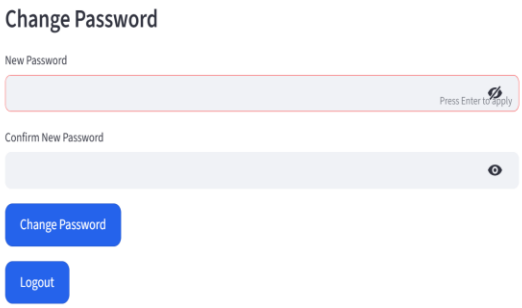
Step-4: Multilingual Interface (English/Hindi)

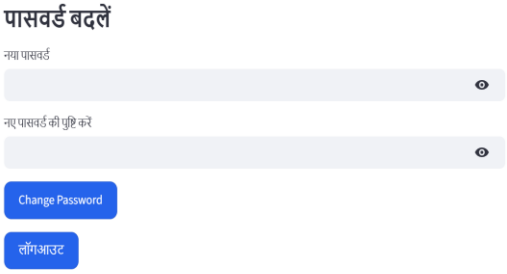
English: -



Hindi: -

Step-5: Change Password





6. Conclusion

The **Wellness Chatbot** is a **secure, user-friendly system** that supports **registration, authentication, and profile management**. It stores details such as **name, age, gender, email, and preferred language** in a **SQLite database**, ensuring **persistence and reliability**. With **multilingual support** in **English and Hindi**, the app improves **accessibility**, while features like **password management** and **profile updates** enhance **usability**. Overall, the project shows how **Streamlit** can be applied beyond **dashboards** to build complete **web applications** with strong **authentication** and **database integration**.