PROJECT REPORT

MOOD JOURNAL

SUBMITTED BY:

ID	NAME
C231523	Nadira Hayet
C231527	Anima Ali Raima
C231544	Tasnim Akter

Table of Contents

Introduction

The Mood Journal Web Application is a full-stack web-based platform that allows users to track their emotional well-being, share anonymous feedback with others, and view public mood updates. The system features secure login and registration, mood posting with optional anonymity, anonymous message replies, and an admin panel to manage user accounts. Designed with both privacy and self-expression in mind, this app promotes emotional awareness and social connection.

Objectives

- To create a mood-tracking system that encourages mental health awareness.
- To allow users to share and receive anonymous messages with optional public visibility.
- To enable secure registration, login, and admin-level user control.
- To demonstrate full-stack development skills using Flask.
- To provide a platform where users can track emotional patterns and interact with others anonymously.

Tools and Technologies Used

Category	Tools/Technologies
Language	Python, HTML, CSS, JavaScript
Framework	Flask (Backend & Routing)

Database	SQLite3
Styling	Bootstrap 4, Font Awesome, Custom CSS
Hosting	Localhost / Flask Dev Server

System Architecture

• Frontend: HTML5, CSS3, Bootstrap for UI/UX.
• Backend: Flask Framework for request handling, user sessions, and database logic.
• Database: SQLite (tables: Users, Moods, AnonymousMessages, Replies).
• User Roles:
- Admin: Can access admin panel, manage users.
- User: Can post moods, send messages, reply to anonymous inputs.

Frontend Development

• Designed responsive UI using Bootstrap and custom CSS.
• Used Jinja2 templating for dynamic HTML rendering.
• Developed dedicated pages:
- Login & Registration
- Dashboard with mood history (calendar-style view)
- Mood input form with public/private toggle
- Anonymous message system
- Public mood and reply feed

• Integrated animated backgrounds and emoji icons for better emotional UX.

Backend Development

• Built using Flask Framework:
- Models: Defined using SQLAlchemy ORM for User, Mood, AnonymousMessage, and Reply. - Routes: Implemented all core functionalities including login, mood posting, reply handling, etc.
- Templating: Used Jinja2 syntax for template inheritance and variable rendering.
- Admin Tools: User management, edit/delete functionalities, and user promotion to admin.

• Security Measures:
- Session management with Flask-Session.
- Input validation for forms.
- Password protection with hashing (can be added using Werkzeug or Flask-Bcrypt).

Features

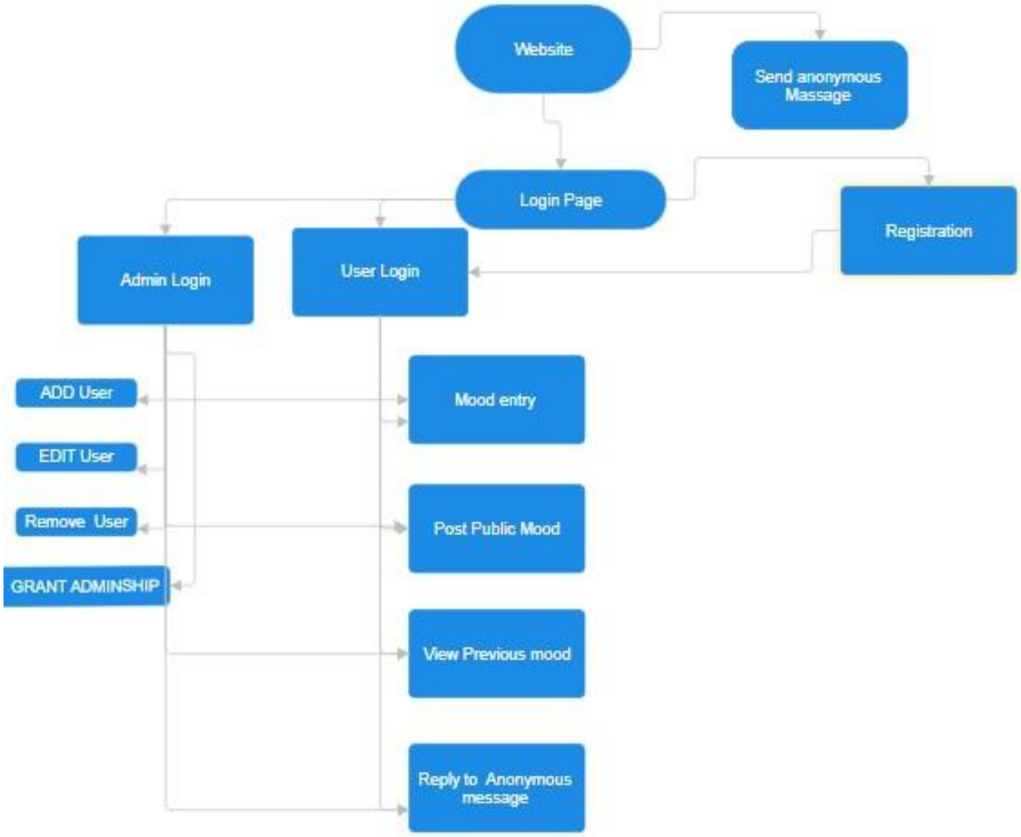
User Features:

- Login / Registration with email and password.
- Post current mood with optional message and visibility toggle (public/private).
- Send anonymous messages to any registered user.
- Reply to received anonymous messages with selected mood type.
- View personalized mood feed and mood history.
- Browse the public feed of moods and replies.

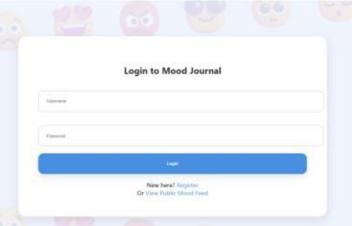
Admin Features:

- Access the /admin dashboard.
- View, edit, and delete users.
- Promote users to admin.
- Add new users from the admin panel.

Flowchart



Screenshot

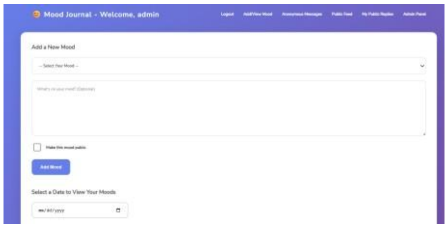


Login to Mood Journal

Username

Password

New here? Register
Or View Public Mood Feed



Mood Journal - Welcome, admin

Add a New Mood

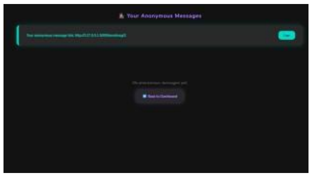
-- Select Your Mood --

What's on your mind? (Optional)

□ Make this mood public

Select a Date to View Your Moods

mm/dd/yyyy



Your Anonymous Messages

Your anonymous message link: http://127.0.0.1:5000/[illegible]

No anonymous messages yet.

Conclusion

This project provided a comprehensive hands-on experience in full-stack web development using Flask. From session handling to relational database design and user role management, the project helped strengthen both backend and frontend capabilities. The Mood Journal App is scalable, secure, and user-friendly. It can be further enhanced by adding features like password hashing, email verification, and analytics for mood tracking.