**Daily Journal App**

**1. Introduction**

**1.1 Purpose**

The purpose of this application is to provide users with a secure, private digital journal where they can write daily entries, track their moods, and receive personalized mental wellness support through an AI-powered chatbot.

**1.2 Scope**

This is a full-stack web application for personal use, featuring user authentication, a journal interface with mood tracking and calendar view, and an AI Mood Assistant chatbot that gives reflective, empathetic responses. Users can view, edit, and manage their journal entries privately.

**1.3 Intended Audience**

* Individuals seeking a mental wellness journaling tool
* Developers or evaluators reviewing portfolio projects
* Mental health support platforms (for future integrations)

**1.4 Technologies Used**

* **Frontend**: React.js, Tailwind CSS
* **Backend**: Node.js, Express.js
* **Database**: MySQL
* **AI**: OpenAI

**2. Overall Description**

**2.1 Product Perspective**

The application is standalone and web-based. It requires internet connectivity to function and relies on third-party services (OpenAI API).

**2.2 Product Functions**

* User registration and login
* Create/edit/delete journal entries
* Mood tracking per entry
* Calendar view of entries
* AI Mood Assistant chatbot
* Save and review chat logs

**2.3 User Classes and Characteristics**

|  |  |
| --- | --- |
| **User Type** | **Description** |
| Regular | Can create/edit/view personal data and chat with AI |

**2.4 Constraints**

* Requires internet for AI/chatbot functionality
* AI Assistant should not replace licensed mental health support
* Must comply with data privacy best practices

**3. System Features**

**3.1 User Authentication**

**Description**: Users must register and log in securely to use the app.  
**Functional Requirements**:

* R1: Users can register with their name, phone number, email, and password
* R2: Login should verify credentials
* R3: Use JWT or session cookies for secure access

**3.2 Journal Entry Management**

**Description**: Users can write, view, edit, and delete journal entries.  
**Functional Requirements**:

* R4: Create new entries with title, content, and mood
* R5: View all previous entries
* R6: Edit or delete existing entries
* R7: Entries are private and user-specific

**3.3 Mood Tracking**

**Description**: Users can select their mood (Happy, Sad, Neutral) for each entry.  
**Functional Requirements**:

* R8: Mood field required during entry creation
* R9: Display mood visually (emoji or icon) in dashboard/calendar

**3.4 Calendar View**

**Description**: Users can view entries based on date  
**Functional Requirements**:

* R10: Show which days have entries
* R11: Clicking a date display that day’s journal entry
* R12: Optionally highlight days by mood

**3.5 AI Mood Assistant**

**Description**: Provides users with emotional support and journaling prompts  
**Functional Requirements**:

* R13: Users can input how they feel in a chatbot
* R14: AI responds with short, empathetic, and reflective messages
* R15: Chat history is stored and user-specific
* R16: Use OpenAI API with prompt tuning to maintain tone

**4. Non-Functional Requirements**

**4.1 Performance**

* Should load all dashboard elements within 2 seconds
* API response time should not exceed 1 second (except AI)

**4.2 Security**

* Passwords must be hashed
* User data must be private
* Use HTTPS and sanitize inputs

**4.3 Usability**

* Should be mobile responsive
* Simple, clean UI inspired by Therachat (calm colors, large touch targets)

**4.4 Maintainability**

* Organized codebase with component separation
* Environment variables used for API keys
* Modular services for AI, auth, and entries

**5. Database Design**

**Tables:**

* users
* journal\_entries
* chat\_logs

**6. External Interface Requirements**

**6.1 OpenAI API**

Used for AI chatbot responses.

* Input: User message
* Output: AI reply
* Endpoint: https://api.openai.com/v1/chat/completions