**MindEase: Habit & Mood Tracker App**

**Course: Mobile Application Development (MAD)**

Instructor: Ma’am Maryam

Submitted by: 22SW021 & 22SW018

**MindEase Project Report**

**1. Real World Problem Identification**

In today’s fast-paced and stressful world, mental well-being has become one of the most important aspects of a healthy lifestyle. However, people often find it difficult to recognize patterns in their moods or emotional states, which prevents them from seeking help at the right time.  
Many individuals struggle to track their daily emotions or mental health progress effectively due to busy schedules and lack of awareness. Without a proper record of mood fluctuations, it becomes challenging to identify triggers of stress, anxiety, or depression.

To address this real-world problem, a solution is required that allows users to monitor their emotions regularly, gain insights into their mental health trends, and promote self-awareness for better well-being.

**2. Proposed Solution**

**MindEase** is a mood tracking mobile application designed to help users understand their emotional patterns over time.  
The app enables users to log their moods daily by selecting an emotion (such as happy, sad, anxious, calm, etc.) and optionally adding notes about their day. Over time, the application visualizes the user’s emotional data to identify trends and help them make informed decisions about their mental health.

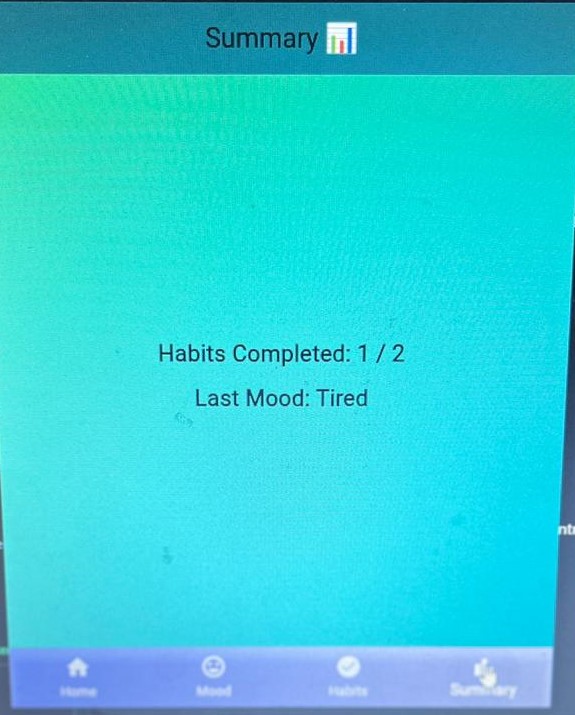
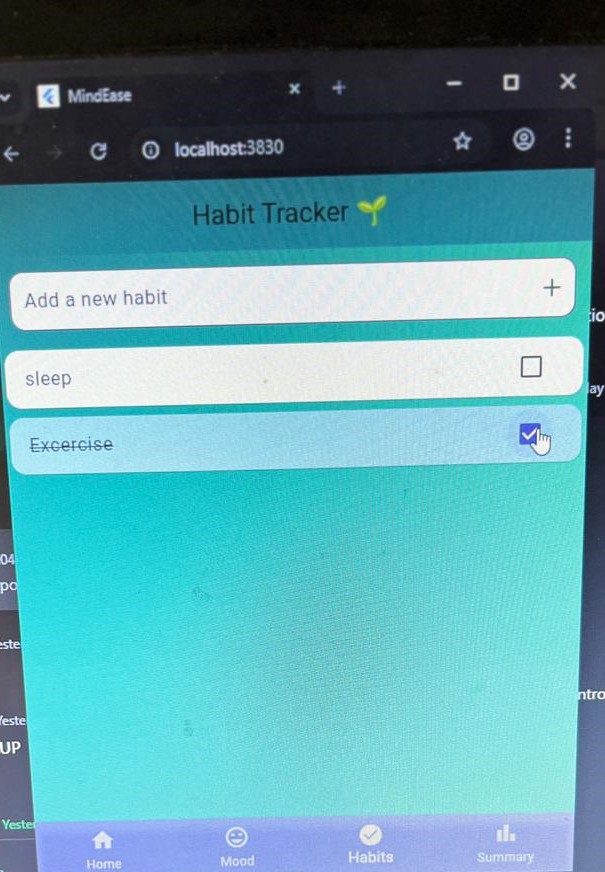
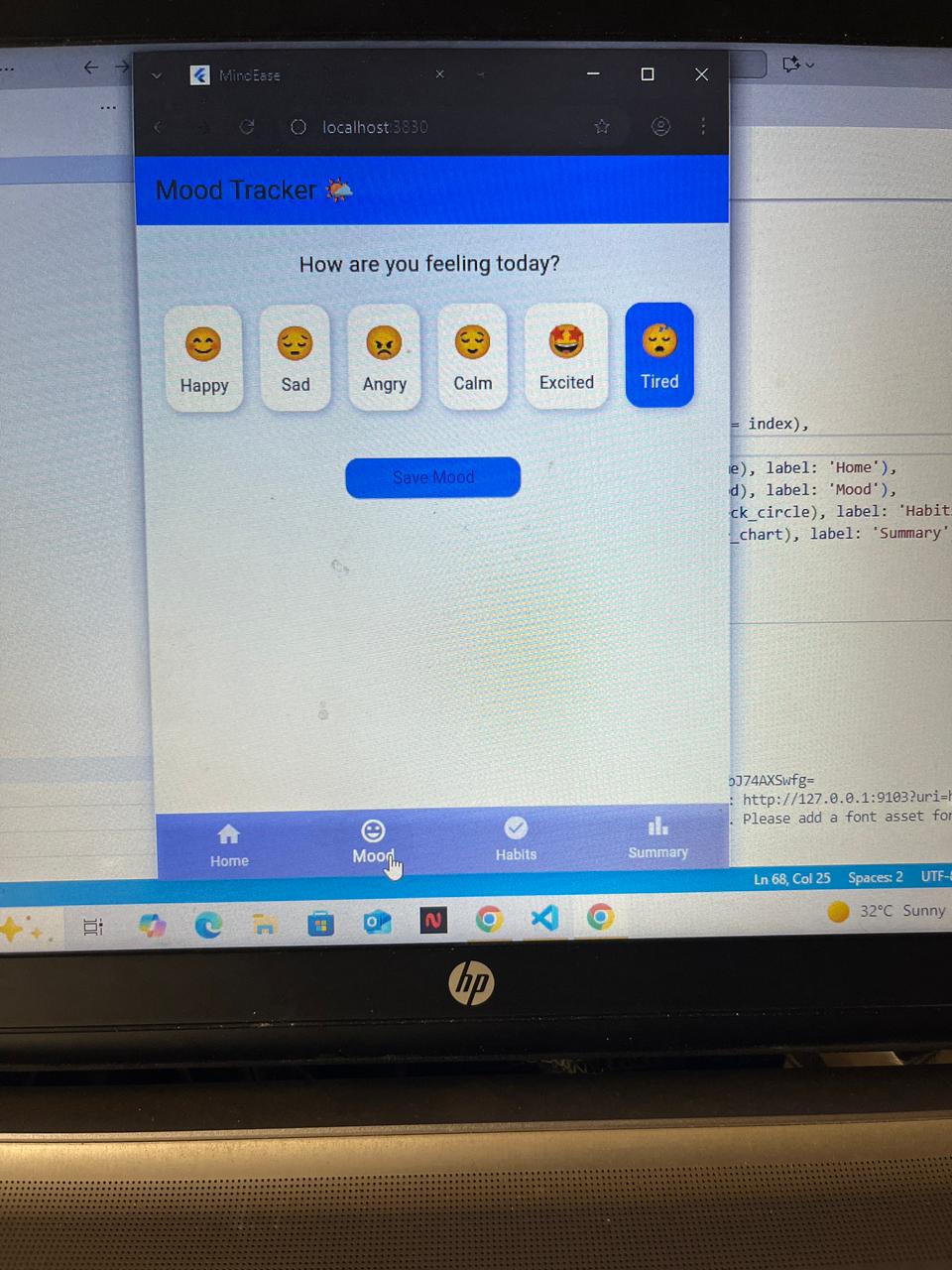
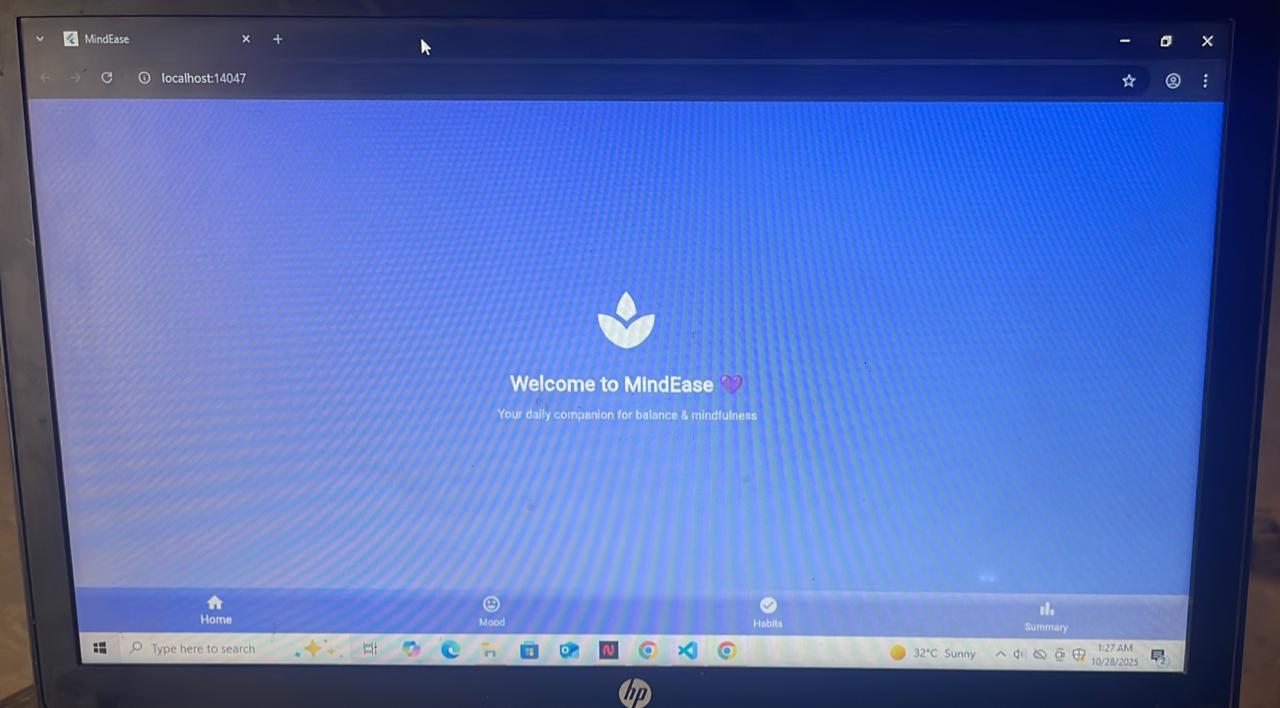
**Key Features:**

* Simple and intuitive interface for mood tracking.
* Daily reminders for users to log their emotions.
* Visualization of mood trends through charts or graphs.
* Local data storage ensuring user privacy.
* Personalized insights to help users identify emotional triggers.

MindEase aims to promote emotional awareness and mental wellness through consistent tracking and reflection.

**3. Responsive User Interfaces**

The app is designed using **Flutter**, ensuring smooth and responsive performance across multiple screen sizes and devices. The user interface follows a clean, minimalistic, and calming design to match the app’s theme of mental wellness.

****

| **Screen Name** | **Description** | **Placeholder** |
| --- | --- | --- |
| Home Screen | Displays current day’s mood and quick access to add a new mood entry. |  |
| Mood Entry Screen | Allows users to choose emotion and write notes. |  |
| Mood History | Shows a list of past moods with timestamps. |  |
| Insights Screen | Visualizes mood patterns over days/weeks. |  |

**4. Data Storage**

MindEase uses a **local database** (such as **SQLite**) to store user data securely within the mobile device.  
This approach was chosen over cloud databases like Firebase because:

* It provides **offline accessibility**, allowing users to track moods without internet connectivity.
* It ensures **data privacy**, as sensitive emotional data remains on the user’s phone.
* It reduces dependency on external servers and avoids storage costs.

The local database stores entries such as:

* Date and Time
* Mood Category (e.g., Happy, Sad, Calm)
* Notes (optional)

This structure supports quick retrieval and visualization of mood trends.

**5. [Optional] APIs/Packages/Plug-ins**

Currently, MindEase operates without external APIs to maintain offline functionality.  
However, several Flutter packages were used to enhance development efficiency and UI design, such as:

* **sqflite** – for local database management.
* **provider** – for state management.
* **charts\_flutter** – for displaying mood trend graphs.

These packages were selected because they are stable, lightweight, and widely supported in the Flutter community.

**6. Issues and Bugs Encountered and Resolved during Development**

During development, several issues were encountered and resolved, as described below:

| **Issue** | **Description** | **Resolution** |
| --- | --- | --- |
| Database Insertion Error | Initial entries were not being stored correctly. | Fixed by restructuring the SQLite query and initializing the database before use. |
| UI Overflow | On smaller screens, mood selection buttons overflowed. | Implemented responsive layouts using Flexible and Expanded widgets. |
| State Management Problem | Mood list was not updating after adding a new entry. | Introduced the Provider package for efficient state updates. |
| App Crash on Restart | The database connection wasn’t closing properly. | Ensured dispose() function handled database closure correctly. |

These improvements helped enhance app stability and user experience.