

BÁO CÁO ĐỒ ÁN MÔN HỌC

Lớp: IT003.CTTN.P21

SINH VIÊN THỰC HIỆN

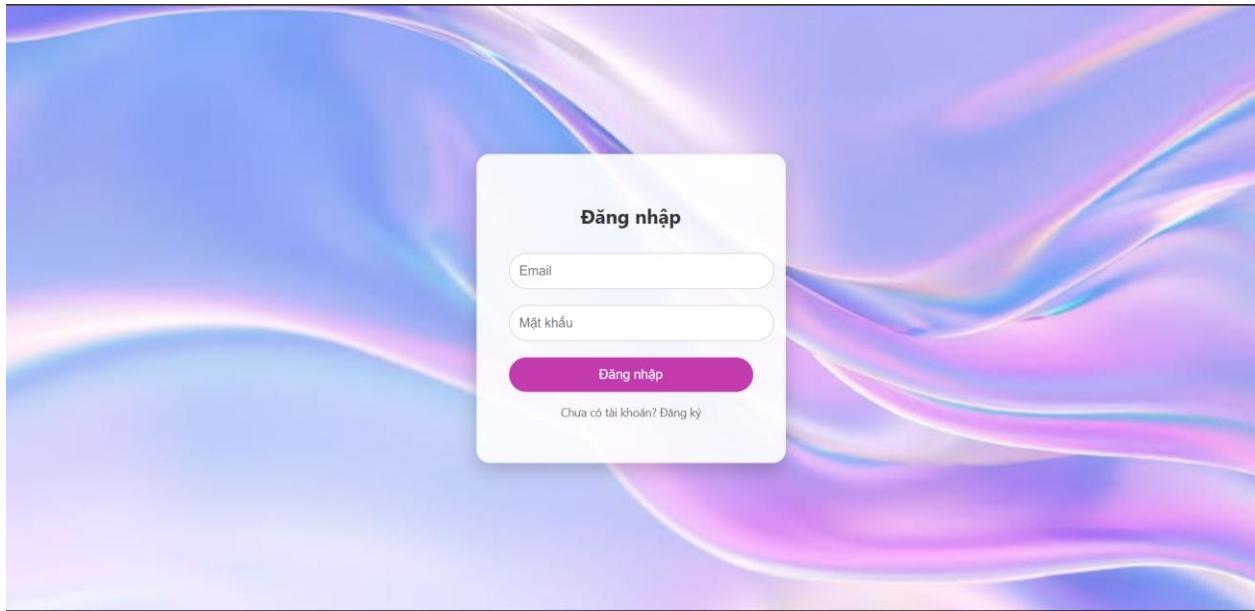
Mã sinh viên: 24520020

Họ và tên: Hà Bùi Trọng Nghĩa

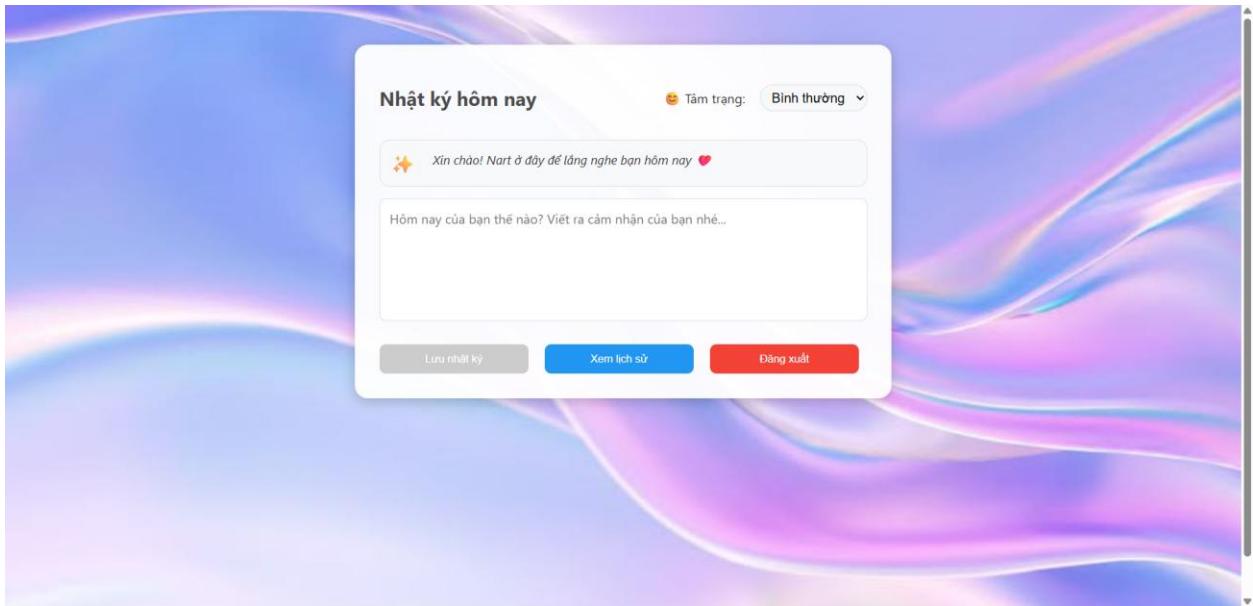
TÊN ĐỀ TÀI: The Life Book – Nhật ký lưu trữ

CÁC NỘI DUNG CẦN BÁO CÁO:

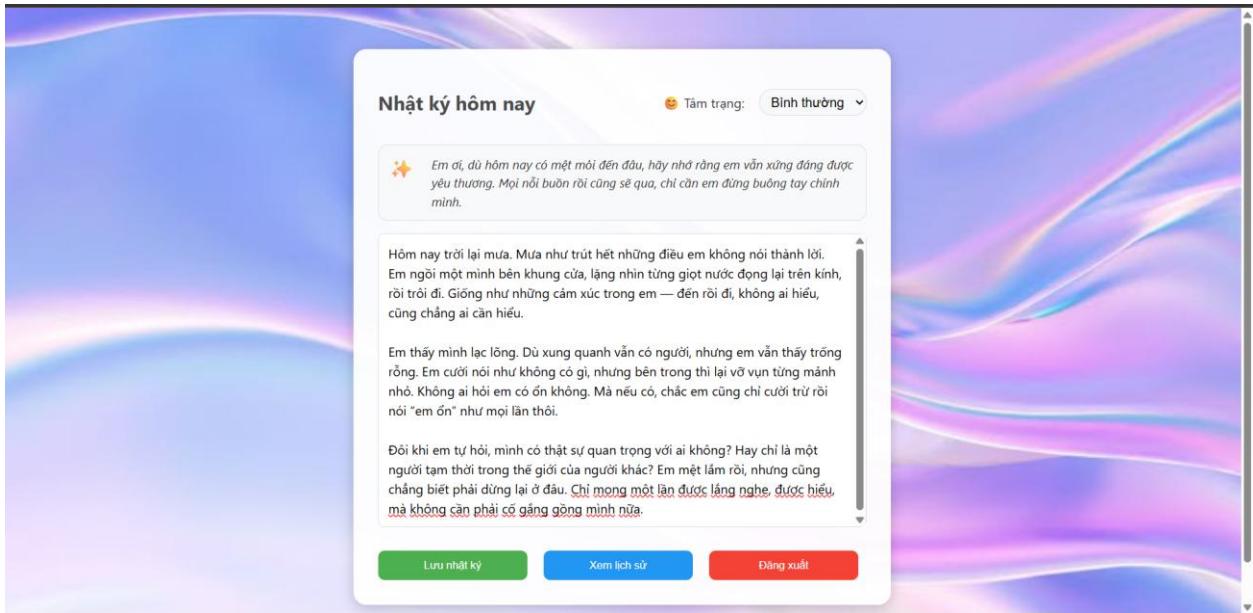
1. Giới thiệu đồ án: Phần mềm là một ứng dụng web dùng để lưu trữ nhật ký và bày tỏ cảm xúc với những gì mình đã viết trong nhật ký.
- Link github: https://github.com/Nartvn/DSA_BTBONUS1
2. Quá trình thực hiện:
 - a. Tuần 1: Lên ý tưởng.
 - b. Tuần 2: Thiết kế trang web.
 - c. Tuần 3: Chỉnh sửa lỗi và cải thiện thuật toán.
 - d. Tuần 4: Tối ưu về mặt hình ảnh và cải thiện thuật toán.
3. Kết quả đạt được: Hoàn thành sản phẩm đáp ứng được các tính năng đề ra.
4. Tài liệu tham khảo:
<https://firebase.google.com/>
<https://ai.google.dev/gemini-api>
<https://chatgpt.com/>
5. Phụ lục 1: Giới thiệu (demo) kết quả
- Trang đăng nhập:



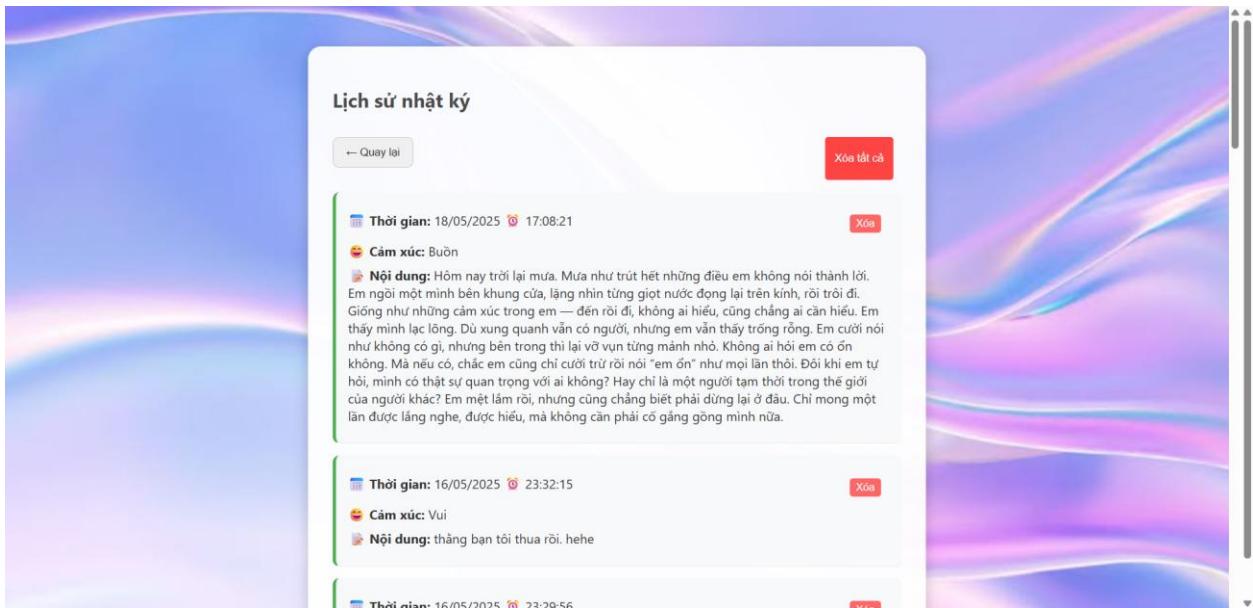
- Có thể tạo tài khoản để có tính cá nhân hóa và riêng tư cho mỗi người.
- Trang viết nhật ký:



- Có thể thấy giao diện mông mơ và bắt mắt khiến người dùng dễ dàng nhận biết và sử dụng. Trong trang viết nhật ký có thể điều chỉnh cảm xúc tùy vào người dùng. Ngoài ra còn có thanh AI Nart chat sẽ tâm sự với người dùng.



- Có thể thấy AI Nart tâm sự với người dùng về đoạn nhật ký họ mới viết, sau đây là trang lưu lại lịch sử nhật ký để người dùng có thể dễ dàng coi lại:



6. Phụ lục 2: docstring

- App.js:

```
/**
```

```
* Component `App`
```

```
*
```

- * Đây là component gốc của ứng dụng, định nghĩa hệ thống định tuyến bằng `react-router-dom`.
- * Ứng dụng sử dụng `` để quản lý các route và điều hướng giữa các trang.
- *
- * Các route được định nghĩa:
 - * - `/` : Trang đăng nhập/đăng ký (component 'Login')
 - * - `/diary` : Trang viết nhật ký (component 'Diary')
 - * - `/chart` : Trang thống kê biểu đồ (component 'ChartPage')
 - * - `/history` : Trang lịch sử các bản ghi nhật ký (component 'History')
 - *
- * Mỗi đường dẫn được gắn với một component tương ứng.
- */

- Login.js:

```
/***
 * Component `Login`
 *
 * Giao diện người dùng để đăng nhập hoặc đăng ký tài khoản thông qua Firebase Authentication.
 *
 * - Sử dụng `useState` để lưu thông tin email và mật khẩu.
 * - Nếu chế độ là "đăng nhập" (`isLogin` là true), gọi `signInWithEmailAndPassword`.
 * - Nếu chế độ là "đăng ký" (`isLogin` là false), gọi `createUserWithEmailAndPassword`.
 * - Sau khi thành công, chuyển hướng người dùng đến trang `/diary`.
 *
 * Các thành phần giao diện gồm:
 * - Ô nhập email
 */


```

- * - Ô nhập mật khẩu
- * - Nút thực hiện đăng nhập/đăng ký
- * - Liên kết chuyển đổi giữa 2 chế độ đăng nhập/đăng ký
- */

- History.js:

```
/***
 * Component `History`
 *
 * Giao diện hiển thị lịch sử các bài viết nhật ký của người dùng hiện tại.
 * Dữ liệu được lấy từ Firestore, nơi lưu trữ các bản ghi có `userId` trùng với tài khoản đã đăng nhập.
 *
 * Chức năng chính:
 * - Hiển thị danh sách các nhật ký (đã được sắp xếp theo thời gian).
 * - Cho phép xóa từng nhật ký riêng lẻ.
 * - Cho phép xóa tất cả nhật ký của người dùng.
 *
 * Sử dụng các hook:
 * - `useState` để lưu danh sách các entry.
 * - `useEffect` để tải dữ liệu khi component được render.
 */
/***
 * Hàm `formatDateTime(dateStr, timeStr)`
 *
 * Hàm định dạng ngày và giờ từ chuỗi `dateStr` và `timeStr` về dạng dễ đọc.
 *
 * @param {string} dateStr - Ngày dạng chuỗi, ví dụ: "2024-05-17" hoặc "17/05/2024"
 * @param {string} timeStr - Giờ dạng chuỗi, ví dụ: "14:30"
 */
```

* @returns { string } - Chuỗi đã định dạng, ví dụ: "17/05/2024 14:30"

*/

- Diary.js:

/**

* Component Diary - Trang nhật ký cá nhân với tích hợp AI Gemini

*

* Cho phép người dùng viết nhật ký hàng ngày với lựa chọn tâm trạng,

* đồng thời tương tác với trợ lý ảo Nart (AI Gemini) để nhận phản hồi

* nhẹ nhàng, đồng cảm và tích cực dựa trên nội dung nhật ký.

*

* Các chức năng chính:

* - Nhập nội dung nhật ký, chọn tâm trạng hiện tại.

* - Gửi nhật ký lên Firestore, lưu kèm thông tin người dùng, ngày giờ.

* - Tự động điều chỉnh chiều cao khung textarea theo nội dung.

* - Gợi ý câu hỏi nhanh giúp người dùng viết nhật ký dễ dàng hơn.

* - Phản hồi tự động từ AI Gemini dựa trên nội dung nhập.

* - Chức năng đăng xuất người dùng.

*

* State:

* - content: nội dung nhật ký.

* - mood: tâm trạng hiện tại.

* - isSubmitting: trạng thái đang gửi dữ liệu.

* - nartState: trạng thái của trợ lý ảo Nart (đang gõ, tin nhắn, hiển thị gợi ý).

*

* Hooks:

* - useEffect để điều chỉnh chiều cao textarea khi nội dung thay đổi.

* - useRef để tham chiếu textarea và form.

* - useNavigate để chuyển hướng trang.

*

* Các hàm:

* - consultNart(userText): gọi API Gemini để nhận phản hồi AI dựa trên nội dung nhật ký.

* - handleSubmit(e): xử lý gửi nhật ký lên Firestore.

* - handleLogout(): đăng xuất người dùng.

*

* @component

* @returns JSX.Element

*/