**Project Progression Form 2**

**Team Name: MinDrift (21102033 Kwon Daehoon, 21102052 Lee Jeongyun, 21102056 Joo Youngjin)**

**Project Title: MinDrift**

**Core Features (Progress)**

* Feature 1: Emotion-based chat with ChatGPT-4o-mini (100% implemented)
* Feature 2: AI-generated diary based on communication history. (100% implemented)
* Feature 3: Archiving, deleting diaries and chatting logs for each user (100% implemented)
* Feature 4: User authentication by JWT (100% implemented)
* Feature 5: Automatic deleting past conversations at 4 a.m. every day to clean up the DB. (100% implemented)

**Database Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Primary Key | Foreign Keys | Field |
| users | id | - | user\_id, username, password, created\_at |
| messages | id | user\_id → users(id) | role, content, created\_at |
| diaries | id | user\_id → users(id) | content, summary, created\_at |

1. users

* Purpose: Stores user account information.
* Description: Contains user ID (user\_id), display name (username), encrypted password (password), and the account creation timestamp (created\_at).

2. messages

* Purpose: Stores daily chat conversation logs between the user and GPT.
* Description: Each message is linked to a user through user\_id, and includes the role (role: either user or assistant), message content, and the timestamp when the message was created.

3. diaries

* Purpose: Stores automatically generated diary entries and summaries.
* Description: Based on the user’s daily chat history, GPT generates a diary entry (chat log) and a summary (diary). Each entry is linked to a user through user\_id, and only one entry is created or updated per date to prevent duplication.

**Backend API Endpoints**

* List all planned API endpoints, grouped by resource. For each endpoint include:

“users” table

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Route** | **Description** | **Progress** |
| POST | /auth/register | Register a new user | 100% |
| POST | /auth/login | Authenticate and get JWT token | 100% |
| GET | /auth/me | Get current authenticated user | 100% |
| PUT | /auth/username | Update username | 100% |

“messages” table

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Route** | **Description** | **Progress** |
| POST | /api/chat | Send a user message and get GPT reply | 100% |
| GET | /api/messages | Get chat history for the user | 100% |

“diaries” table

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Route** | **Description** | **Progress** |
| POST | /api/diary | Generate diary from raw input text | 100% |
| POST | /api/diary/from-history | Generate diary from chat history | 100% |
| GET | /api/diary/archive | Get diary and messages for a specific date | 100% |
| GET | /api/diary/dates | Get list of all diary entry dates | 100% |
| GET | /api/diary/id-by-date | Get diary ID for a specific date | 100% |
| DELETE | /api/diary/:id | Delete a diary entry by ID | 100% |

**Front-end progress**

Briefly describe your progress on the front-end UI:

• Have you started building components/pages?

- Mostly done

• Which tools/libraries are you using? (e.g., plain HTML/CSS, Bootstrap, React, etc.):

- React (with Next.js), Tailwind CSS, TypeScript, Lucide Icons(in Archive page)

• Key front-end pages/components completed so far: (e.g. Home, Dashboard, Login, Signup, etc.) You may attach screenshots.

- Login Page

- Signup Page

- Chat Page

- Archive Page (Diary List & Details)

- Username Edit Feature

- JWT-based Authentication (Token stored in localStorage)