API Documentation: Gemini Health Intelligence Platform

Version: 1.0.0 **Status:** Live **Contact:** [ishaan406061@gmail.com] **Base URL:** `https://hsc1606.onrender.com`

1. Overview

The **Gemini Health Intelligence Platform** is a state-of-the-art backend service engineered to deliver AI-driven medical symptom analysis. It provides a secure, scalable, and robust API that leverages Google's Gemini 2.5 Flash for advanced multimodal (text and image) analysis, integrates real-time geolocation for healthcare provider recommendations, and ensures data persistence and user privacy through a secure authentication and storage layer.

This document provides a comprehensive guide for developers to integrate with and utilize the full capabilities of the API.

2. Authentication

Authentication is managed via **JWT (JSON Web Tokens)** using an `OAuth2PasswordBearer` flow. All endpoints, excluding `/signup` and `/login`, require a valid Bearer Token to be included in the `Authorization` header of the request.

Workflow:

1. A new user is created via the `POST /signup` endpoint.
2. The user authenticates using the `POST /login` endpoint, providing their email (`username`) and `password`.
3. A successful login returns an `access_token`.
4. This `access_token` must be prefixed with `Bearer` and included in the `Authorization` header for all subsequent protected requests.

Example Header: `Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...`

3. API Endpoints

User Management

`POST /signup`

Creates a new user account in the system.

- **URL:** `/signup`
- **Method:** `POST`
- **Request Body:** `application/json`

Field	Type	Description
`name`	string	The full name of the user.
`email`	string	The user's unique email address.
`password`	string	The user's chosen password.

Example Request:

```
{
  "name": "Aisha Khan",
  "email": "aisha.khan@example.com",
  "password": "MySecurePassword@2025"
}
```

Success Response (`200 OK`):

```
{
  "id": 1,
  "name": "Aisha Khan",
  "email": "aisha.khan@example.com"
}
```

`POST /login`

Authenticates an existing user and returns a JWT access token.

- **URL:** `/login`
- **Method:** `POST`
- **Request Body:** `application/x-www-form-urlencoded`

Field	Type	Description
`username`	string	The user's email address.
`password`	string	The user's plain-text password.

Success Response (`200 OK`):

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJhaXNoYS5raGFuQGV4YW1wbGUuY29tIiwi
ZXhwIjoxNzYwNjU1MzQxfQ.abcdefg...",
  "token_type": "bearer"
}
```

Core Functionality

`POST /analyze/text`

Analyzes a user's symptoms described in text and returns a structured analysis and nearby hospital recommendations.

- **URL:** `/analyze/text`
- **Method:** `POST`
- **Authentication:** `Bearer Token` required.
- **Request Body:** `application/json`

Field	Type	Description
`symptoms`	string	A detailed description of the user's symptoms.
`latitude`	float	(Optional) The user's latitude.
`longitude`	float	(Optional) The user's longitude.

Example Request:

```
{
  "symptoms": "Experiencing a dull, persistent headache and dizziness for the past 4 hours.",
  "latitude": 12.9716,
  "longitude": 77.5946
}
```

Success Response (`200 OK`): A JSON object containing the Gemini analysis. The result is also saved to the user's history.

`POST /analyze/image`

Analyzes symptoms from an uploaded image and optional accompanying text.

- **URL:** `/analyze/image`
- **Method:** `POST`
- **Authentication:** `Bearer Token` required.
- **Request Body:** `multipart/form-data`

Field	Type	Description
`image`	file	The image file of the symptom.
`symptoms`	string	(Optional) Accompanying text description.
`latitude`	float	(Optional) The user's latitude.

Field	Type	Description
`longitude`	float	(Optional) The user's longitude.

Success Response (`200 OK`): A JSON object containing the multimodal Gemini analysis. The result and image URL are saved to the user's history.

`GET /history`

Retrieves the complete query history for the authenticated user, ordered by most recent first.

- **URL:** `/history`
- **Method:** `GET`
- **Authentication:** `Bearer Token` required.

Success Response (`200 OK`): A JSON array where each object is a past query record.

```
[
    {
        "id": 2,
        "created_at": "2025-10-17T12:30:00.123Z",
        "user_id": 1,
        "symptom_text": "This rash appeared on my arm...",
        "image_url":
"https://<...>.supabase.co/storage/v1/object/public/symptom_images/1/abc-123.jpg",
        "response_data": {
            "condition": "Contact Dermatitis",
            "confidence_score": "High",
            // ... other analysis fields
        }
    },
    {
        // ... previous history entry
    }
]
```

---

4. Data Models

Gemini Response Schema (`response_data`)

All analysis endpoints return a structured JSON object with the following potential fields.

Field	Type	Description
`condition`	string	The most probable medical condition.
`confidence_score`	string	"Low", "Medium", or "High".
`description`	string	A detailed explanation of the condition.

Field	Type	Description
`recommended_steps`	array	A list of actionable next steps for the user.
`disclaimer`	string	A mandatory medical disclaimer.
`nearby_hospitals`	array	(If location provided) A list of nearby hospitals.

5. Error Handling

The API uses standard HTTP status codes to indicate the success or failure of a request.

Code	Meaning	Possible Reason
`200`	**OK**	The request was successful.
`400`	**Bad Request**	Invalid request body or missing required fields.
`401`	**Unauthorized**	Missing, invalid, or expired JWT access token.
`422`	**Unprocessable Entity**	The request was well-formed but semantically incorrect.
`500`	**Internal Server Error**	An unexpected error occurred on the server side.

6. Disclaimer

This API is a proof-of-concept and is **not a substitute for professional medical advice**. The analysis provided is generated by an AI model and should be used for informational purposes only. Always consult a qualified healthcare professional for diagnosis and treatment.