My APP API GUIDE

I. Auth module - Endpoints

1. User Registration (Sign-Up)

- Endpoint: /api/auth/register
- HTTP Method: POST
- Description: Allows a new user to register by providing their details
- Case to consider: How should the register with the gmail?
- Request Body:

```
{
    "name": "Namecinho",
    "lastName": "Rivera Rivera",
    "birthday": "02-02-2001",
    "password": "123ABC",
    "phoneNumber": "978525987" || This will work like the username
}
```

- Response:

```
{
    "message": "User registered successfully",
    "userId": "12345" || We need the userId in the response?
}
```

2. User Login (Sign-In)

- Endpoint: /api/auth/login
- HTTP Method: POST
- Description: Authenticates the user and returns a JWT if the credentials are valid.
- Case to consider: How should the login with the gmail?
- Request Body:

```
{
    "phoneNumber": "978525987",
    "password": "securepassword123"
}
```

- Response:

```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

3. User Logout (Optional)

- Endpoint: /api/auth/logout
- HTTP Method: POST
- Description: Invalidates the user's JWT.
- Request Body:

```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

- Response:

```
{
    "message": "Logout successful"
}
```

4. Password Reset – Send Recovery Code

- Endpoint: /api/auth/password-reset/initiate
- HTTP Method: POST
- Description: Initiates password reset by sending a 6-digit code via SMS to the user’s registered phone number.
- Note: You’ll need a Twilio, Vonage, or AWS SNS number to send SMS.
- Request Body:

```
{
    "phoneNumber": "987654321"
}
```

- Response:

```
{
    "message": "Recovery code sent to +51987654321",
    "expiresIn": 300  // Code expires in 5 minutes.
}
```

5. Password Reset – Confirm Code

- Endpoint: /api/auth/password-reset/verify
- HTTP Method: POST
- Description: Validates the 6-digit recovery code sent to the user.
- Notes:
  - Generate a short-lived JWT (resetToken) to authorize the password update in Step 3.
  - Store the code securely (hashed) in your DB and compare it here.
- Request Body:

```
{
    "phoneNumber": "987654321",
    "recoveryCode": "123456"  // 6-digit code
}
```

- Response:

```
{
    "message": "Code verified successfully",
    "resetToken": "xyz789"  // Temporary token for Step 3
}
```

6. Password Reset – Update Password

- Endpoint: /api/auth/password-reset/complete
- HTTP Method: POST
- Description: Updates the user's password after code verification.
- Note: Invalidate resetToken immediately after use.
- Request Body:

```
{
    "resetToken": "xyz789",  // From Step 2
    "newPassword": "NewSecure123!"  // Enforce complexity rules
}
```

- Response:

```
{
    "message": "Password updated successfully"
}
```

7. Update Password

- Endpoint: /api/auth/update-password
- HTTP Method: POST || PATCH
- Description: Allows the user to update their password
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "currentPassword": "oldpassword123",
    "newPassword": "newpassword456"
}
```

- Response:

```
{
    "message": "Password updated successfully"
}
```

II. User module - Endpoints

1. Get User Profile

- Endpoint: /api/users/me
- HTTP Method: GET
- Description: Returns the authenticated user's profile information.
- Headers: `Authorization : JWT`
- Question: Can we get the user information just with the JWT?
- Response:

```
{
    "userId": "user01"
    "name": "Namecinho",
    "lastName": "Rivera Rivera",
    "birthday": "02-02-2001",
    "password": "123ABC",
    "phoneNumber": "978525987"
}
```

2. Update User Information

- Endpoint: /api/users/me
- HTTP Method: PUT
- Description: Allows the user to update their profile information (e.g., email, username, firstName, lastName).
- Headers: `Authorization : JWT`

- Request Body:

```
{
    "name": "Namecinho",
    "lastName": "Rivera Rivera",
    "birthday": "02-02-2001",
    "phoneNumber": "978525987",
    "avatarImg": "https://example.com/uploads/avatars/user123.jpg"
}
```

- Response:

```
{
    "message": "User updated successfully"
}
```

3. Delete User Account

- Endpoint: /api/users/me
- HTTP Method: DELETE
- Description: Deletes the authenticated user's account. || Just a logical delete
- Headers: `Authorization : JWT`
- Response:

```
{
    "message": "User deleted successfully"
}
```

4. Get All Users (Admin Only)

- Endpoint: /api/users
- HTTP Method: GET
- Description: Returns a list of all users (paginated).
- Headers: `Authorization : JWT`
- Query Parameters (optional):
  - page = 1 (default: 1)
  - limit = 10 (default: 10)
- Response:

```
{
    "users": [
        {
            "userId": "user01",
            "name": "Namecinho",
```

```
            "lastName": "Rivera Rivera",
            "phoneNumber": "978525987",
            "isActive": true
        },
        {
            "userId": "user02",
            "name": "Jane",
            "lastName": "Doe",
            "phoneNumber": "987654321",
            "isActive": false
        }
    ],
    "total": 2,
    "page": 1,
    "limit": 10
}
```

5. Get User by ID (Admin Only)

- Endpoint: /api/users/{userId}
- HTTP Method: GET
- Description: Returns details of a specific user by ID.
- Headers: `Authorization : JWT`
- Query Parameters (optional):
  - page = 1 (default: 1)
  - limit = 10 (default: 10)
- Response:

```
{
    "userId": "user01",
    "name": "Namecinho",
    "lastName": "Rivera Rivera",
    "birthday": "02-02-2001",
    "phoneNumber": "978525987",
    "isActive": true,
    "createdAt": "2023-10-20T12:00:00Z"
}
```

6. Update User by ID (Admin Only)

- Endpoint: /api/users/{userId}
- HTTP Method: PUT
- Description: Updates a user's details by ID (admin-only).
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "name": "Updated Name",
    "lastName": "Updated LastName",
    "phoneNumber": "987654321",
    "isActive": false
}
```

- Response:

```
{
    "message": "User updated successfully",
    "userId": "user01"
}
```

7. Delete User by ID (Admin Only)

- Endpoint: /api/users/{userId}
- HTTP Method: DELETE
- Description: Logical delete of a user by ID (sets isActive: false).
- Headers: `Authorization : JWT`
- Response:

```
{
    "message": "User deactivated successfully",
    "userId": "user01"
}
```

III. Confidence Constacs module - Endpoints

1. Get Confidence Contacts

- Endpoint: /api/users/me/confidence-contacts
- HTTP Method: POST
- Description: Adds a new confidence contact for the authenticated user.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "firstName": "John",
    "lastName": "Doe",
    "phoneNumber": "+1234567890",
    //"isActive": true || by default is true
}
```

- Response:

```
{
    "message": "Confidence contact added successfully",
    "contactId": "12345",
    "firstName": "John",
    "lastName": "Doe",
    "phoneNumber": "+1234567890",
    "isActive": true
}
```

2. Get All Confidence Contacts

- Endpoint: /api/users/me/confidence-contacts
- HTTP Method: GET
- Description: Retrieves all confidence contacts for the authenticated user.
- Headers: `Authorization : JWT`
- Response:

```
[
    {
        "contactId": "12345",
        "firstName": "John",
        "lastName": "Doe",
        "phoneNumber": "+1234567890",
        "isActive": true
    },
    {
        "contactId": "67890",
        "firstName": "Jane",
        "lastName": "Smith",
        "phoneNumber": "+0987654321",
        "isActive": false
    }
]
```

3. Get a Specific Confidence Contact

- Endpoint: /api/users/me/confidence-contacts/{contactId}
- HTTP Method: GET
- Description: Retrieves a specific confidence contact by its ID.
- Headers: `Authorization : JWT`
- Response:

```
{
    "contactId": "12345",
    "firstName": "John",
```

```
    "lastName": "Doe",
    "phoneNumber": "+1234567890",
    "isActive": true
}
```

4. Update a Confidence Contact

- Endpoint: /api/users/me/confidence-contacts/{contactId}
- HTTP Method: PUT
- Description: Updates the details of a specific confidence contact.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "firstName": "Johnny",
    "lastName": "Doe",
    "phoneNumber": "+1234567890",
    "isActive": false
}
```

- Response:

```
{
    "message": "Confidence contact updated successfully",
    "contactId": "12345",
    "firstName": "Johnny",
    "lastName": "Doe",
    "phoneNumber": "+1234567890",
    "isActive": false
}
```

5. Delete a Confidence Contact

- Endpoint: /api/users/me/confidence-contacts/{contactId}
- HTTP Method: DELETE
- Description: Deletes a specific confidence contact.
- Headers: `Authorization : JWT`
- Response:

```
{
    "message": "Confidence contact deleted successfully"
}
```

6. Change the Active State of a Confidence Contact

- Endpoint: /api/users/me/confidence-contacts/{contactId}/active
- HTTP Method: PATCH
- Description: Updates the isActive state of a confidence contact.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "isActive": false
}
```

- Response:

```
{
    "message": "Confidence contact state updated successfully",
    "contactId": "12345",
    "isActive": false
}
```

IV. Tracking module - Endpoints

1. Start Tracking

- Endpoint: /api/tracking/start
- HTTP Method: POST
- Description: Initiates a tracking session for the user's journey.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "userId": "12345",
    "origin":{
        "latitude": 45.2128,
        "longitude": -60.0230
    }
    "destination": {
        "latitude": 40.7128,
        "longitude": -74.0060
    }
}
```

- Response:

```
{
    "trackingId": "track-789",
```

```
    "message": "Tracking started",
    "checkpointInterval": 2  // Seconds (configurable),
    "route": "Idk what we will receive here, but supose that is the route
defined by the external API".
}
```

2. Update Live Coordinates

- Endpoint: /api/tracking/update
- HTTP Method: POST
- Description: Sends the user's current coordinates at fixed intervals.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "trackingId": "track-789",
    "coordinates": {
        "latitude": 40.7128,
        "longitude": -74.0060,
        "timestamp": "2023-10-20T12:00:00Z"
    }
}
```

- On Route Response:

```
{
    "status": "ON_ROUTE",
    "nextCheckIn": 2
}
```

- Off Route Response:

```
{
    "status": "OFF_ROUTE",
    "lastValidPoint": { "latitude": 40.7120, "longitude": -74.0050 },
    "deviationRadius": 15.23  // Meters
}
```

3. Deviation Handling. User Prompt ("Are you okay?")

- Check Deviation Radius: Logic. Server calculates if the user is outside the 50m(defined radius) radius.
    - **If inside radius:** No action (continue monitoring).
    - **If outside radius:** Trigger a prompt (next endpoint).
- Endpoint: /api/tracking/prompt-response

- HTTP Method: POST
- Description: Handles the user's response to the safety prompt.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "trackingId": "track-789",
    "response": "NO",  // "YES" or "NO"
    "coordinates": { // Current location
        "latitude": 40.7128,
        "longitude": -74.0060
    }
}
```

- NO Response: // Internally send the SOS message to her/his active confidence contacts

```
{
    "message": "SOS triggered",
    "contactsNotified": ["contact-1", "contact-2"],
    "lastCoordinates": { "latitude": 40.7128, "longitude": -74.0060 }
}
```

- YES Response: // Internally recalibrate the route.

```
{
    "message": "Recalibrating route",
    "newRouteId": "route-abc",
    "nextCheckIn": 2,
    "route": "Idk what we will receive here, but supose that is the route
defined by the external API".
}
```

- Timeout Response: // The server automatically triggers SOS if no response is received.

4. SOS Signal to Contacts

- Endpoint: /api/emergency/sos
- HTTP Method: POST
- Description: Sends real-time coordinates to trusted contacts.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "trackingId": "track-789",
    "userId": "12345",
```

```
    "coordinates": { //Optional
        "latitude": 40.7128,
        "longitude": -74.0060
    }
}
```

- Response:

```
{
    "message": "SOS sent to contacts",
    "fallbackUsed": false  // True if offline and used last known
coordinates
}
```

5. Arrival Confirmation

- Endpoint: /api/tracking/arrived
- HTTP Method: POST
- Description: Notifies the server that the user reached the destination safely.
- Headers: `Authorization : JWT`
- Request Body:

```
{
    "trackingId": "track-789",
    "userId": "12345",
    "coordinates": {
        "latitude": 40.7128,
        "longitude": -74.0060
    }
}
```

- Response:

```
{
    "message": "Arrival confirmed",
    "contactsNotified": ["contact-1", "contact-2"]
}
```

6. Get Tracking History

- Endpoint: /api/tracking/history?userId=12345
- HTTP Method: GET
- Headers: `Authorization : JWT`
- Response:

```
{
    "path": [
        {
            "latitude": 40.7128,
            "longitude": -74.0060,
            "timestamp": "2023-10-20T12:00:00Z"
        },
        {
            "latitude": 40.7130,
            "longitude": -74.0050,
            "timestamp": "2023-10-20T12:00:02Z"
        }
    ]
}
```