

# API Documentation

## Environment-wise API Response Time

Environment	Ideal Response Time	Acceptable Response Time	Slow Response Time
Localhost	< 100ms	100ms - 300ms	> 500ms
Hosted (Production)	< 200ms	200ms - 500ms	> 500ms

## API Response Time Breakdown

API Type	Expected Response Time
Static Response (No DB, Minimal Logic)	50 - 150ms
Database Queries (Optimized, Cached)	100 - 300ms
Complex Processing (Multiple Queries, External APIs)	300 - 700ms
Heavy Computation / Large Data Processing	700ms - 1s+

## 1.User Signup API

Method: POST

Endpoint: /auth/signup

Description: Allows users to sign up by providing necessary details such as name, email, mobile number, date of birth, gender, and password.

Headers:

```
{
  "Content-Type": "application/json"
}
```

Request Body (JSON):

```
{
  "name": "Jane Doe",
  "email": "janedoe@example.com",
  "mobile_no": "9877548210",
  "dob": "1998-08-22",
  "gender": 1,
  "password": "password123"
}
```

### Response Body (JSON):

```
{
  "message": "User registered successfully!",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

### Signup API Possible Errors:

- **400 Bad Request:** "All fields are required!" → Occurs when any required field (name, email, mobile\_no, dob, gender, password) is missing.
- **400 Bad Request:** "Mobile number must be exactly 10 digits!" → Happens if the mobile number is not in the correct 10-digit format.
- **400 Bad Request:** "Gender must be either (Male) or (Female)!" → Triggered when the gender value is not **0** (Female) or **1** (Male).
- **400 Bad Request:** "Password must be at least 8 characters long!" → Raised when the password length is less than 8 characters.
- **500 Internal Server Error:** "Database error occurred!" → Happens when there's an issue inserting data into the database.
- **500 Internal Server Error:** "Internal Server Error" → Occurs due to unexpected errors in the backend.

## 2. User Login

**Method:** POST

**Endpoint:** /auth/login

**Description:** Allows users to log in using email and password.

**Headers:**

```
{  
  "Content-Type": "application/json"  
}
```

**Request Body (JSON):**

```
{  
  "email": "janedoe@example.com",  
  "password": "password123"  
}
```

**Response Body (JSON):**

```
{  
  "message": "Login successful!",  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

**Login API Possible Errors:**

- **400 Bad Request:** "Email and password are required!" → Triggered when either the email or password field is missing in the request body.

- **400 Bad Request:** "Password must be at least 8 characters long!" → Happens if the password is shorter than 8 characters.
- **401 Unauthorized:** "Email not found or account does not exist" → Raised when the provided email is not found in the database.
- **401 Unauthorized:** "Incorrect password" → Occurs when the entered password does not match the stored hashed password.
- **500 Internal Server Error:** "Database error occurred!" → Happens if there's an issue retrieving user data from the database.
- **500 Internal Server Error:** "Internal Server Error" → Triggered due to unexpected errors in the backend.

### 3. User Profile API

**Method:** GET

**Endpoint:** /auth/profile

**Description:** Fetches the user's profile details.

**Headers:**

```
{  
  "Authorization": "Bearer <your_access_token>"  
}
```

**Response Body (JSON):**

```
{
  "message": "User profile fetched successfully!",
  "profile": {
    "id": 1,
    "name": "Jane Doe",
    "email": "janedoe@example.com",
    "mobile_no": "9877548210",
    "dob": "1998-08-22",
    "gender": "Female",
    "age": 26,
    "profile_photo":
    "..."
  }
}
```

#### Profile API Possible Errors:

- **401 Unauthorized:** "Missing or invalid token" → Occurs when the request does not include the Authorization header or the token is invalid.
- **404 Not Found:** "User not found" → Happens if the user ID from the token does not match any record in the database.
- **500 Internal Server Error:** "Database error occurred!" → Raised when there's an issue fetching user profile data from the database.
- **500 Internal Server Error:** "Internal Server Error" → Occurs due to unexpected backend issues.

## 4. Get User Info API (Age & Gender)

Method: GET

Endpoint: /auth/user-info

**Description:** Fetches the user's age and gender.

**Headers:**

```
{  
  "Authorization": "Bearer <your_access_token>"  
}
```

**Response Body (JSON):**

```
{  
  "message": "User info fetched successfully!",  
  "age": 26,  
  "gender": "Female"  
}
```

**User Info API Possible Errors:**

- **401 Unauthorized:** "Missing or invalid token" → Occurs if the request does not include the Authorization header or the token is invalid.
- **404 Not Found:** "User not found!" → Triggered when no user exists in the database for the given userId.
- **500 Internal Server Error:** "Database error occurred!" → Raised if there's an issue retrieving user information from the database.
- **500 Internal Server Error:** "Internal Server Error" → Happens due to unexpected backend errors.

## 5. User Profile Update API

**Method:** PUT

**Endpoint:** /auth/update-profile

**Description:** Allows users to update their profile details.

**Headers:**

```
{  
  "Authorization": "Bearer <your_access_token>",  
  "Content-Type": "application/json"  
}
```

### Request Body (JSON):

```
{  
  "name": "Jane Doe",  
  "mobile_no": "9876543210",  
  "gender": 1,  
  "dob": "1998-08-22",  
  "password": "password123",  
  "profile_photo": "/9j/4AAQSkZJRgABAQAAQ..."  
}
```

### Response Body (JSON):

```
{  
  "message": "Profile updated successfully!"  
}
```

### Update Profile API Possible Errors:

- **400 Bad Request:** "Password is required to update profile!" → Occurs when the password field is missing in the request body.
- **400 Bad Request:** "Mobile number must be 10 digits!" → Happens if the provided mobile number is not exactly 10 digits.

- **400 Bad Request:** "Gender must be 0 (Female) or 1 (Male)!" → Triggered if the gender value is invalid.
- **400 Bad Request:** "No fields to update!" → Raised when no updatable fields (name, mobile\_no, gender, dob, image) are provided in the request.
- **401 Unauthorized:** "Incorrect password!" → Occurs when the provided password does not match the stored password.
- **401 Unauthorized:** "Missing or invalid token" → Happens if the request does not include the Authorization header or the token is invalid.
- **404 Not Found:** "User not found!" → Triggered if no user exists in the database for the given userId.
- **500 Internal Server Error:** "Database error occurred!" → Raised when there's an issue updating user profile data in the database.
- **500 Internal Server Error:** "Internal Server Error" → Happens due to unexpected backend errors.

## 6.AI Prediction API

**Method:** POST

**Endpoint:** /ai

**Description:** Processes an array of 13 numerical features and returns a predicted class.

**Headers:**

```
{  
  "Authorization": "Bearer <your_access_token>",  
  "Content-Type": "application/json"  
}
```

**Request Body (JSON):**



```
{  
  "features": [63, 0, 3, 145, 233, 1, 0, 150, 0, 2.3, 0, 0, 1]  
}
```

### Response Body (JSON):

```
{  
  "success": true,  
  "predicted_class": 1  
}
```

### Heart Disease Prediction API Possible Errors:

- **400 Bad Request:** "Invalid input format. 'features' must be an array." → Occurs when the request body does not contain a valid array under the features key.
- **400 Bad Request:** "Invalid input. 'features' array must contain exactly 13 values." → Happens if the features array does not contain exactly 13 numerical values.
- **400 Bad Request:** "Invalid value at index X. Expected a number between Y and Z." → Triggered when any feature value is out of the allowed range or is not a number.
- **401 Unauthorized:** "Missing or invalid token" → Occurs when the request does not include the Authorization header or the token is invalid.
- **500 Internal Server Error:** "Prediction or Database error" → Raised when there's an issue calling the AI model or storing prediction results in the database.
- **500 Internal Server Error:** "Database error" → Happens if an error occurs while inserting prediction data into the database.

## 7.AI Prediction History API

**Method:** GET

**Endpoint:** /ai/history

**Description:** Fetches a user's past AI predictions.

**Headers:**

```
{  
  "Authorization": "Bearer <your_access_token>"  
}
```

**Response Body (JSON):**

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "uid": 1,  
      "age": 63,  
      "sex": 0,  
      "cp": 3,  
      "trestbps": 145,  
      "chol": 233,  
      "fbs": 1,  
      "restecg": 0,  
      "thalach": 150,  
      "exang": 0,  
      "oldpeak": 2.3,  
      "slope": 0,  
      "ca": 0,  
      "thal": 1,  
      "target_multi": 1,  
      "created_at": "2025-03-24T12:00:00Z"  
    }  
  ]  
}
```

**History API Possible Errors:**

- **401 Unauthorized:** "Unauthorized: User ID missing in token" → Occurs when the request does not include a valid token or the user ID is missing from the token.
- **500 Internal Server Error:** "Database error" → Happens when there's an issue retrieving the history data from the database.
- **500 Internal Server Error:** "Server error" → Triggered due to an unexpected backend failure.