

# Health Check Pro

Health Check Pro is a user-friendly health website built using the MERN stack

---

## Authentication

Contains all Apis related to authentication

---

### POST Login User

`http://localhost:5000/api/user/login`

### API Request Description

This endpoint is used to authenticate a user and obtain a login session. The request should be sent as an HTTP POST to the specified URL with the user's email and password provided in the request body.

### Request Body

- email (string, required): The email address of the user.
- password (string, required): The password of the user.

### API Response

The response for this request is a JSON object conforming to the following schema:

json

```
{
  "token": "string",
  "user": {
    "username": {
      "type": "string"
    },
    "email": {
      "type": "string"
    },
    "profilePic": {
      "type": "string",

```

The response contains the user's ID, a token for authentication, and the expiration date of the token.

## Body raw (json)

---

json

```
{
  "email": "ashuvidya2003com",
  "password": "#Ash12345"
}
```

## POST Register User

http://localhost:5000/api/user/signup

## User Signup

This endpoint allows users to sign up by providing their username, email, age, gender, password, and profile picture.

### Request Body

- `username` (string): The username of the user.
- `email` (string): The email address of the user.
- `age` (number): The age of the user.
- `gender` (string): The gender of the user.
- `password` (string): The password for the user account.
- `confirmPassword` (string): The confirmation of the password.
- `profilePic` (string): The profile picture of the user.

Example:

json

```
{
  "username": "Ash12345",
  "email": "ash12345@gmail.com",
  "age": 22,
  "gender": "Male",
  "password": "#Pass12345",
  "confirmPassword": "#Pass12345",
  "profilePic": ""
}
```

## Response

- `token` (string): The authentication token for the user.
- `email` (string): The email address of the user.
- `username` (string): The username of the user.
- `profilePic` (string): The profile picture of the user.

Example:

json

```
{
  "token": "",
  "email": "",
  "username": "",
  "profilePic": ""
}
```

## Body raw (json)

json

```
{
  "username": "Ash12345",
  "email": "ash12345@gmail.com",
  "age": 22,
  "gender": "Male",
  "password": "#Pass12345",
  "confirmPassword": "#Pass12345",
  "profilePic": ""
}
```

## POST Verify the Sent OTP

<http://localhost:5000/api/user/verify-otp>

The endpoint `/api/user/verify-otp` is a POST request used to verify a user's OTP (One-Time Password) for a specific email address.

## Request Body

- `email` (string, required): The email address for which the OTP is to be verified.
- `otp` (string, required): The OTP to be verified.

## Response

The response of this request is a JSON object with the following schema:

json

```
{
  "message": "OTP is successfully verified!"
}
```

The `status` property indicates the status of the OTP verification, and the `message` property provides additional information about the verification status.

**Body** raw (json)

---

json

```
{
  "email": "ashuvaidya2003.com",
  "otp": 234234
}
```

---

## POST Send OTP to reset password

<http://localhost:5000/api/user/send-otp>

### Send OTP to User

This endpoint is used to send a one-time password (OTP) to the user's email for verification.

#### Request Body

- email (string, required): The email address to which the OTP will be sent.

#### Response

The response will include the status of the OTP sending process.

**Body** raw (json)

---

json

```
{
  "message": "OTP is successfully verified!"
}
```

```
}
```

```
"email": "j154u.test@inbox.testmail.app"
```

## POST Reset Password

http://localhost:5000/api/user/reset-password

### Reset Password

This endpoint is used to reset the password for a user account.

#### Request Body

- `email` (string): The email address of the user.
- `password` (string): The new password for the user account.
- `confirmPassword` (string): The confirmation of the new password.

#### Response

The response will indicate the success or failure of the password reset request.

#### Body raw (json)

json

```
{
  "email": "ashuvaidya2003@gmail.com",
  "password": "#Pass212345",
  "confirmPassword": "#Pass212345"
}
```

## POST Admin Signup



http://localhost:5000/api/user/signupAdmin

#### AUTHORIZATION Bearer Token

Token <token>

#### Body raw (json)

json

```
{
  "username": "Ash12345",
  "email": "ashish@hcapro.com",
  "password": "#Pass12345",
  "confirmPassword": "#Pass12345"
}
```

---

## Assessment

Contains all operations occurring for assessment section of the app

---

### GET Start a new Round

#### API Request Description

This endpoint allows the user to start a new round of assessment. The request should be sent via an HTTP POST method to the specified URL with the user's ID included in the request body.

#### Request Body

- user (string, required): The ID of the user initiating the new assessment round.

#### API Response

The API responds with a status code of 400 and a JSON object with a "message" property. The "message" property may contain additional information about the error encountered during the request.

#### JSON Schema for Response

json

```
{
  "type": "object",
  "properties": {
    "message": {
      "type": "string"
    }
  }
}
```

## GET Fetch User Assessment Status

http://localhost:5000/api/assessment/status

### GET /api/assessment/status

This endpoint retrieves the status of the assessment for a specific user.

#### Request

- Method: GET
- URL: http://localhost:5000/api/assessment/status
- Headers:
  - Content-Type: application/json
- { "userId": "671ea4beb96f9974ea865716" }

#### Response

The response is in JSON format and follows the schema below:

json

```
{
  "type": "object",
  "properties": {
    "attemptNumber": {
      "type": "integer"
    },
    "completedCategories": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  }
}
```

- Status: 200
- Content-Type: application/json
- { "attemptNumber": 0, "completedCategories": [] }

#### Related Responses

- [ { "\_id": "", "categoryId": "", "questionText": "", "options": [], "selectedOption": null, "createdAt": "", "updatedAt": "", "\_\_v": 0 } ]
- [ { "\_id": "", "categoryName": "", "description": "", "createdAt": "", "updatedAt": "", "\_\_v": 0 } ]
- { "message": "" }
- [ { "\_id": "", "categoryId": "", "questionText": "", "options": [], "selectedOption": null, "createdAt": "", "updatedAt": "", "\_\_v": 0 } ]

Body raw (json)

---

json

```
{
  "userId": "671ea4beb96f9974ea865716"
}
```

---

## GET Get all Categories

http://localhost:5000/api/category

The API endpoint retrieves a list of assessment categories.

The response of this request can be documented as a JSON schema:

json

```
[
  {
    "_id": "string",
    "categoryName": "string",
    "description": "string",
    "createdAt": "string",
    "updatedAt": "string",
    "__v": 0
  }
]
```

---

## GET Get questions of a Specific Category

http://localhost:5000/api/category/67237629366a67da4eb44ba9/questions/

This HTTP GET request retrieves assessment questions by ID from the API. The request should be made to `http://localhost:5000/api/assessment/questions/{categoryid}`.

### Request

No request body is required for this endpoint.

### Response

The response will have a status code of 200 and a content type of application/json. The body of the response will contain an array of objects, each representing an assessment question. Each question object will have the following properties:



an array of objects, each representing an assessment question. Each question object will have the following properties:

- `_id` : The unique identifier for the question.
- `categoryId` : The category ID to which the question belongs.
- `questionText` : The text of the assessment question.
- `options` : An array of options for the question.
- `selectedOption` : The selected option for the question, if any.
- `createdAt` : The timestamp of when the question was created.
- `updatedAt` : The timestamp of when the question was last updated.
- `__v` : Version control field.

Please note that the actual values for these properties will be present in the response, but they have been intentionally masked for privacy reasons.

---

## PATCH Submit Test



`http://localhost:5000/api/assessment/submit`

The API endpoint allows users to submit assessment data via an HTTP PATCH request to `http://localhost:5000/api/assessment/submit`. The response of this request can be documented as a JSON schema as follows:

```
json
```

```
{
  "type": "object",
  "properties": {
    "message": {
      "type": "string"
    }
  }
}
```

## AUTHORIZATION Bearer Token

Token	<token>
-------	---------

---

## GET Fetch most recent Attempt

This endpoint makes an HTTP GET request to retrieve the latest incomplete assessment attempt. The response will include details such as the user ID, attempt number, assessments, overall score, completion status, and timestamps.

### Request Body

This endpoint does not require a request body.

## Response

The response will have a status code of 200 and a content type of application/json. The response body will contain the details of the latest incomplete assessment attempt, including the user ID, attempt number, assessments, overall score, completion status, and timestamps.

## Related Responses

This endpoint's related responses from other API calls include data models for assessment categories, questions, and messages, with status codes of 200 and 201.

---

## GET Get all the attempts



`http://localhost:5000/api/assessment/all-attempts`

The endpoint retrieves all assessment attempts. The response is in JSON format and includes an array of objects, each representing an assessment attempt. Each assessment attempt object contains the following properties:

- `"_id"`: The unique identifier of the assessment attempt
- `"userId"`: The identifier of the user associated with the attempt
- `"attemptNumber"`: The number of the attempt
- `"assessments"`: An array of assessments
- `"overallScore"`: The overall score of the attempt
- `"isComplete"`: Indicates whether the attempt is complete
- `"date"`: The date of the attempt
- `"createdAt"`: The creation date of the attempt
- `"updatedAt"`: The date of the last update
- `"__v"`: Version control field

Additionally, related responses from other endpoints are provided for reference, with their respective status codes. The data models of these related responses are largely similar to the main response.

json

```
{
  "allAttempts": [
    {
      "_id": "",
      "userId": "",
      "attemptNumber": 0,
      "assessments": [],
      "overallScore": 0,
      "isComplete": true,
      "date": "",
      "createdAt": ""
    }
  ]
}
```

Token

&lt;token&gt;

## GET Get attempt by Id



http://localhost:5000/api/assessment/attempt/6742fc446565a4ba0a808498

The endpoint retrieves the assessment attempt with the specified ID.

The response for this request can be documented as a JSON schema:

json

```
{
  "type": "object",
  "properties": {
    "_id": { "type": "string" },
    "userId": { "type": "string" },
    "attemptNumber": { "type": "integer" },
    "overallScore": { "type": "integer" },
    "overallMaxScore": { "type": "integer" },
    "isComplete": { "type": "boolean" },
    "date": { "type": "string" },
    "createdAt": { "type": "string" }
  }
}
```

Token

&lt;token&gt;

## GET Get assessment from attempt



http://localhost:5000/api/assessment/attempt/6742fc446565a4ba0a808498/672f913fd6d08dbd4cc03a39

The API endpoint retrieves the assessment attempt category details for a specific user of specific attempt.

The response body follows the JSON schema below:

json

```
{
  "userResponse": {
    "type": "object",
    "properties": {

```

```
properties: {  
  "categoryId": {  
    "type": "string"  
  },  
  "questions": {  
    "type": "array",  
    "items": {  
      "type": "object"  
    }  
  }  
}
```

The related response from other endpoints of this API includes a leaderboard array with objects containing username and score. The response status code for this request is 200.

## AUTHORIZATION Bearer Token

Token	<token>
-------	---------

## Admin

Contains CRUD operations for managing assessment section

## POST Add a Categories



http://localhost:5000/api/admin/add-category

This endpoint allows you to add a new category for assessment. The HTTP POST request should be sent to http://localhost:5000/api/assessment/add-category with the following payload in the raw request body type:

json

```
{  
  "categoryName": "Lifestyle",  
  "description": "This category covers various aspects of lifestyle assessments."  
}
```

The `categoryName` parameter specifies the name of the category, while the `description` parameter provides a brief description of the category being added.

Upon a successful request, the response will include the details of the newly added category.

## AUTHORIZATION Bearer Token

Token	<token>
-------	---------

**Body** raw (json)

json

```
{
  "categoryName": "Lifestyle",
  "description": "This category covers various aspects of daily living that contribute to overall well-being."
}
```

## POST Add a question for category



http://localhost:5000/api/admin/add-question

### Add Question

This endpoint allows you to add a new question to the assessment for the category.

#### Request Body

- `categoryId` (string): The ID of the category to which the question belongs.
- `totalScore` (number): The total score assigned to the question.
- `questionText` (string): The text of the question.
- `options` (array of objects): An array containing the options for the question, where each object has:
  - `optionText` (string): The text of the option.
  - `score` (number): The score associated with the option.
- `selectedOption` (string, optional): The ID of the selected option.

#### Response

The response for this request will be in JSON format and will adhere to the following schema:

json

```
{
  "type": "object",
  "properties": {
    "status": {
      "type": "string"
    },
    "message": {
      "type": "string"
    },
    "data": {
      "type": "object",

```

Token <token>

Body raw (json)

json

```
{
  "categoryId": "67237629366a67da4eb44ba9",
  "totalScore": 25,
  "questionText": "How many servings of fruits and vegetables do you eat per day?",
  "options": [
    {
      "optionText": "Less than 1 serving",
      "score": 1
    },
    {
      "optionText": "1 serving",
      "score": 2
    },
    {
      "optionText": "2 servings",
      "score": 3
    },
    {
      "optionText": "3-4 servings",
      "score": 4
    },
    {
      "optionText": "5 or more servings",
      "score": 5
    }
  ],
  "selectedOption": null
}
```

## DELETE Delete a Question



<http://localhost:5000/api/admin/delete-question/67237a58fed5102ccde3e5c6>

The API endpoint `DELETE /api/assessment/delete-question/{questionId}` is used to delete a specific question from the assessment.

## Response

The response for this request is a JSON object with the following schema:

json

```
{
  "type": "object",
  "properties": {
    "message": {
      "type": "string"
    }
  }
}
```

The response contains a `message` property of type string, which may provide information about the success or failure of the delete operation.

### AUTHORIZATION Bearer Token

Token	<token>
-------	---------

## PUT Modify a Question



<http://localhost:5000/api/admin/modify-question/67237a36fed5102ccde3e5ba>

## Modify Question

This endpoint allows the modification of a specific question in the assessment.

### Request Body

- `questionText` (string): The modified text of the question.
- `options` (array): An array of options for the question.

### Response

Upon successful modification, the endpoint returns a status code of 200 with an empty JSON response.

### AUTHORIZATION Bearer Token

Token	<token>
-------	---------

**Body** raw (json)

json

```
{  
  "questionText": "This question is modified by modify-question API",  
  "options": []  
}
```

---

## GET Get all Categories

`http://localhost:5000/api/category`

The API endpoint retrieves a list of assessment categories.

---

## GET Get questions of a Specific Category

`http://localhost:5000/api/category/67237629366a67da4eb44ba9/questions`

This HTTP GET request retrieves assessment questions by ID from the API. The request should be made to `http://localhost:5000/api/admin/questions/{categoryid}`.

### Request

No request body is required for this endpoint.

Requires `categoryId` as a parameter

### Response

The response will have a status code of 200 and a content type of `application/json`. The body of the response will contain an array of objects, each representing an assessment question. Each question object will have the following properties:

- `_id` : The unique identifier for the question.
- `categoryId` : The category ID to which the question belongs.
- `questionText` : The text of the assessment question.
- `options` : An array of options for the question.
- `selectedOption` : The selected option for the question, if any.
- `createdAt` : The timestamp of when the question was created.
- `updatedAt` : The timestamp of when the question was last updated.
- `__v` : Version control field.

Please note that the actual values for these properties will be present in the response, but they have been intentionally masked for privacy reasons.



# Leaderboard

Fetches all users and sort them based on scores based on categories

## GET Get Leaderboard Data by Category



http://localhost:5000/api/leaderboard/672f913fd6d08dbd4cc03a39

### API Request Description

This endpoint makes an HTTP GET request to retrieve the leaderboard data for a specific category.

### Response

The response will be in JSON format and will contain an array of leaderboard objects with the following fields:

- `leaderboard` (array)
  - `_id` (string)
  - `userId` (string)
  - `username` (string)
  - `score` (number)
  - `category` (string)
  - `__v` (number)
  - `createdAt` (string)
  - `updatedAt` (string)

Example response:

json

```
{
  "leaderboard": [
    {
      "_id": "",
      "userId": "",
      "username": "",
      "score": 0,
      "category": "",
      "__v": 0,
      "createdAt": "",
      "updatedAt": ""
    }
  ]
}
```

The response will also include a status code of 200 upon successful retrieval of the leaderboard data.

Token

&lt;token&gt;

## GET Fetch Leaderboard by total score



http://localhost:5000/api/leaderboard/

### API Request Description

This endpoint retrieves the leaderboard data via an HTTP GET request to http://localhost:5000/api/leaderboard/. Gets aggregate of all category scores.

### Request Body

This request does not require a request body.

### Response Body

The response will contain a JSON object with a "leaderboard" array. Each item in the "leaderboard" array includes a "username" and a "score" field.

### JSON Schema for Response

json

```
{
  "type": "object",
  "properties": {
    "leaderboard": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "username": {
            "type": "string"
          }
        }
      }
    }
  }
}
```

Token

&lt;token&gt;