

API Specification: Quiz Application

- Shreyansh Agrawal

Successful response format:

Field	Type	Description	Example
<i>code</i>	Integer	HTTP status code	200
<i>data</i>	Object	Response data	[{"name": "John"}]
<i>message</i>	String	Message for end user	"Request Successful"
<i>status</i>	String	HTTP status message	"Ok"

Example:

```
{
  "code": <status code>,
  "data": <response data>,
  "message": <message for end users>,
  "status": <response status>
}
```

Unsuccessful response format:

Field	Type	Description	Example
<i>code</i>	Integer	HTTP status code	401
<i>message</i>	String	Message for end user	"Please Login"
<i>status</i>	String	HTTP status message	"Unauthorized"

Example:

```
{
  "code": <status code>,
  "message": <error message for end users>,
  "status": <response status>
}
```

Notes:

- None of the response fields are optional
- Bearer Token in the Authorization Header is the access token obtained in login request
- Tables and examples under "Request Body" refer to the data that should go in the request body
- Tables and examples under "Response" refer to the data and the message that is received in the "data" and "message" fields of the successful response format respectively as shown above.

1. POST /register

- **Description:** Register a new user
- **Headers:** None
- **Parameters:** None
- **Request Body:**

Field	Type	Description	Example	Required
<i>name</i>	String	User's name	"John Wick"	Yes
<i>email</i>	String	User's email address	"john@example.com"	Yes
<i>username</i>	String	User's username	"john_wick"	Yes
<i>password</i>	String	User's password	"secretpassword"	Yes

Example:

```
{
  "name": "John Wick",
  "email": "john@example.com",
  "username": "john_wick",
  "password": "secretpassword",
}
```

- **Response:**
 - **Success:**
 - Status Code: 201 Created
 - Body: A success message

Field	Type	Description	Example
<i>message</i>	String	A success message	Successfully registered

Example:

```
{
  "message": "Successfully registered"
}
```

- **Error:**
 - Status Code: 400 Bad Request, 409 Conflict, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

2. POST /login

- **Description:** Login an existing user
- **Headers:** None
- **Parameters:** None
- **Request Body:**

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>	<i>Required</i>
<i>username</i>	String	User's username	"john_doe"	Yes
<i>password</i>	String	User's password	"secretpassword"	Yes

Example:

```
{  
  "username": "john_wick",  
  "password": "secretpassword",  
}
```

- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: An access-token and a refresh token

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>access_token</i>	String	Access token	NA
<i>refresh_token</i>	String	Refresh token	NA
<i>message</i>	String	A success message	Logged in successfully

Example:

```
{  
  "access_token": "<access-token>",  
  "refresh_token": "<refresh-token>",  
  "message": "Logged in successfully"  
}
```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

3. POST /logout

- **Description:** Logout a logged in user
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Request Body:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Successfully logged out

Example:

```
{  
  "message": "Successfully logged out"  
}
```

- **Error:**
 - Status Code: 401 Unauthorized, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

4. POST /refresh

- **Description:** Get a refresh token
- **Headers:**
 - Authorization: <Bearer token> (refresh token, required)
- **Parameters:** None
- **Request Body:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: An access-token

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>access_token</i>	String	An access token	NA

Example:

```
{  
  "access_token": "<access-token>"  
}
```

- **Error:**
 - Status Code: 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

5. GET /profile/me

- **Description:** View profile
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Request Body:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: User data

Field	Type	Description	Example
<i>username</i>	String	User's username	john_wick
<i>name</i>	String	User's name	John Wick
<i>email</i>	String	User's email address	john@gmail.com
<i>registration_date</i>	String	Date of registration	2023-12-07

Example:

```
{  
  "username": "john_wick",  
  "name": "John Wick",  
  "email": "john@gmail.com",  
  "registration_date": "2023-12-07"  
}
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

6. PUT /profile/me

- **Description:** Update user profile
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Request Body:**

Field	Type	Description	Example	Required
name	String	Updated name	"John Doe"	Yes
email	String	Updated email	"john@gmail.com"	Yes
username	String	Updated username	"john_doe"	Yes

Example:

```
{
  "name": "John Doe",
  "email": "john@gmail.com",
  "username": "john_wick"
}
```

- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

Field	Type	Description	Example
message	String	A success message	Updated successfully

Example:

```
{
  "message": "Updated successfully"
}
```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

7. PUT /profile/password/me

- **Description:** Update user password
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Request Body:**

Field	Type	Description	Example	Required
<i>current_password</i>	String	Current password	"secretpassword"	Yes
<i>new_password</i>	String	Updated password	"newsecret"	Yes

Example:

```
{  
  "current_password": "secretpassword",  
  "new_password": "newsecret"  
}
```

- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

Field	Type	Description	Example
<i>message</i>	String	A success message	Updated successfully

Example:

```
{  
  "message": "Updated successfully"  
}
```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

8. GET /admins

- **Description:** Get all admin details
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: Response containing a list of admin details

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>user_id</i>	String	User's id	A3cTnU
<i>name</i>	String	User's name	Mathew Murdock
<i>email</i>	String	User's email address	matty@gmail.com
<i>registration_date</i>	String	Date of registration	2024-01-01

Example:

```
[
  {
    "user_id": "A3cTnU",
    "name": "Mathew Murdock",
    "email": "matty@gmail.com",
    "registration_date": 2024-01-15
  },
  {
    "user_id": "A4zqNx",
    "name": "Wilson Fisk",
    "email": "fisk@gmail.com",
    "registration_date": 2023-04-01
  }
]
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

9. POST /admins

- **Description:** Create a new admin account
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Request Body:**

Field	Type	Description	Example	Required
<i>name</i>	String	Admin's name	"Mathew Murdock"	Yes
<i>email</i>	String	Admin's email address	"matty@example.com"	Yes
<i>username</i>	String	Admin's username	"matty"	Yes

Example:

```
{
  "name": "Mathew Murdock",
  "email": "matty@example.com",
  "username": "matty"
}
```

- **Response:**
 - **Success:**
 - Status Code: 201 Created
 - Body: A success message

Field	Type	Description	Example
<i>message</i>	String	A success message	Admin created successfully

Example:

```
{
  "message": "Admin created successfully"
}
```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 409 Conflict, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

10. DELETE /admins/{admin_id}

- **Description:** Delete an existing admin
- **Headers:**
 - Authorization: **<Bearer token>** (required)
- **Parameters:**
 - admin_id: String (Path parameter, ID of the admin to delete)
- **Request Body:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Admin deleted successfully

Example:

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

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

11. GET /players

- **Description:** Get all player details
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: (Response containing a list of player details)

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>user_id</i>	String	User's id	A3cTnU
<i>name</i>	String	User's name	Mathew Murdock
<i>email</i>	String	User's email address	matty@gmail.com
<i>registration_date</i>	String	Date of registration	2024-01-01

Example:

```
[
  {
    "user_id": "P3cTnU",
    "name": "John Wick",
    "email": "john@gmail.com",
    "registration_date": 2024-01-15
  },
  {
    "user_id": "P4zqNx",
    "name": "Charlie Cox",
    "email": "charlie@gmail.com",
    "registration_date": 2024-01-01
  }
]
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

12. DELETE /players/{player_id}

- **Description:** Delete an existing player
- **Headers:**
 - Authorization: **<Bearer token>** (required)
- **Parameters:**
 - player_id: String (Path parameter, ID of the player to delete)
- **Request Body:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Player deleted successfully

Example:

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

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

13. GET /categories

- **Description:** Get all categories details
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: (Response containing a list of categories)

Field	Type	Description	Example
<i>category_id</i>	String	ID of the category	C3cTnU
<i>admin_id</i>	String	ID of the category creator	Ax2VjP
<i>admin_username</i>	String	Username of the creator	matty
<i>category_name</i>	String	Name of the category	REST API

Example:

```
[
  {
    "category_id": "C3cTnU",
    "admin_id": "A3qZnR",
    "admin_username": "matty",
    "category_name": "REST API"
  },
  {
    "category_id": "A4zqNx",
    "admin_id": "A4zqNx",
    "admin_username": "fisk",
    "category_name": "Python"
  }
]
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

14. POST /categories

- **Description:** Create a new category
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Request Body:**

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>	<i>Required</i>
<i>category_name</i>	String	Category name	"Science"	Yes

Example:

```
{  
  "category_name": "Science"  
}
```

- **Response:**
 - **Success:**
 - Status Code: 201 Created
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Category created successfully

Example:

```
{  
  "message": "Category created successfully"  
}
```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 409 Conflict, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

15. PUT /categories/{category_id}

- **Description:** Update an existing category
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:**
 - category_id: String (Path parameter, ID of the category to update)
- **Request Body:**

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>	<i>Required</i>
<i>updated_category_name</i>	String	Updated category name	"Mathematics"	Yes

Example:

```
{  
  "updated_category_name": "Mathematics"  
}
```

- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: Response containing the updated category details

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Category updated successfully

Example:

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

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

16. DELETE /categories/{category_id}

- **Description:** Delete an existing category
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:**
 - category_id: String (Path parameter, ID of the category to delete)
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Category deleted successfully

Example:

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

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

17. POST /categories/{category_id}/questions

- **Description:** Create a question in a specified category
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:**
 - category_id: String (Path parameter, ID of the category to add a question)
- **Request Body:**

Field	Type	Description	Example	Required
question	String	New question text	"What is the capital of France?"	Yes
type	String	MCQ, One Word, t/f	"mcq"	Yes
answer	String	Correct answer	"Paris"	Yes
other_options	Array[String]	List of wrong options	["x", "y", "z"]	Yes

Example:

```
{
  "question": "What is the capital of France?",
  "type": "MCQ",
  "answer": "Paris",
  "other_options": ["Marseille", "Lyon", "Bordeaux"]
}
```

- **Response:**
 - **Success:**
 - Status Code: 201 Created
 - Body: A success message

Field	Type	Description	Example
message	String	A success message	Question created successfully

Example:

```
{
  "message": "Question created successfully"
}
```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 409 Conflict, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

18. **GET** /categories/questions?category_id={category_id}

- **Description:** Get all questions in a specified category if query parameter specified else all questions across all categories
- **Headers:**
 - Authorization: **<Bearer token>** (required)
- **Parameters:**
 - category_id: String (Query parameter, ID of the category to retrieve questions for)
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: Response containing a list of questions for the specified category or list of questions across all categories

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>quiz_data</i>	Array	An array of objects containing quiz data.	See example
<i>category_id</i>	String	ID of the category	"C3cTnU"
<i>category</i>	String	Category name	"Science"
<i>created_by</i>	String	ID of creator of the category	"AP1qsz"
<i>question_data</i>	Array	Array of question objects within the category	See example
<i>question_id</i>	String	ID of the question	"Qe4ZTL"
<i>question_text</i>	String	Text of the question	"Chemical symbol of water is?"
<i>question_type</i>	String	Type of the question	"ONE WORD"
<i>created_by</i>	String	ID of creator of the question	"Az1Ysq"
<i>options</i>	Object	Object containing answer options	See example
<i>answer</i>	String	Correct answer	"H2O"
<i>other_options</i>	Array	Array of other answer options	[]

Example:

```
{
  "quiz_data": [
    {
      "category_id": "C3cTnU",
      "category": "Science",
      "created_by": "AP1qsz",
      "question_data": [
        {
          "question_id": "Qe4ZTL",
          "question_text": "Chemical symbol of water is?",
          "question_type": "ONE WORD",
          "created_by": "Az1Ysq",
          "options": {
            "answer": "H2O",
            "other_options": []
          }
        }
      ]
    },
    {
      "category_id": "C0cQnZ",
      "category": "Maths",
      "created_by": "AP1qsz",
      "question_data": [
        {
          "question_id": "Qe4ZTL",
          "question_text": "Sum of angles of a triangle is?",
          "question_type": "ONE WORD",
          "created_by": "Az1Ysq",
          "options": {
            "answer": "180",
            "other_options": []
          }
        },
        {
          "question_id": "Qe4ZTL",
          "question_text": "Which gas causes greenhouse effect?",
          "question_type": "MCQ",
          "created_by": "Az1Ysq",
          "options": {
            "answer": "Carbon Dioxide",
            "other_options": ["Oxygen", "Nitrogen", "Methane"]
          }
        }
      ]
    }
  ]
}
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

19. POST /categories/questions

- **Description:** Post quiz data including questions, categories, and options
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Request Body:**

Field	Type	Description	Example	Required
quiz_data	Array	An array of objects containing quiz data.	See example	Yes
category	String	The category of the quiz	"Science"	Yes
question_data	Array	An array of objects containing question data within a category.	See example	Yes
question_text	String	The text of the quiz question.	"What is the chemical symbol for water?"	Yes
question_type	String	The type of the question (e.g., "one word", "t/f", "mcq").	"one word"	Yes
options	Object	An object containing options for the question.	See example	Yes
answer	String	The correct answer to the question.	"H2O"	Yes
other_options	Array	An array containing other options for the question (for multiple-choice questions).	["Oxygen", "Nitrogen", "Methane"]	No

Example:

```
{
  "quiz_data": [
    {
      "category": "Science",
      "question_data": [
        {
          "question_text": "What is the chemical symbol for water?",
          "question_type": "one word",
          "options": {
            "answer": "H2O",
            "other_options": []
          }
        },
        {
          "question_text": "What is the chemical symbol for water?",
          "question_type": "t/f",
          "options": {
            "answer": "False",
            "other_options": []
          }
        }
      ]
    }
  ]
}
```

```

    }
  }
],
},
{
  "category": "Maths",
  "question_data": [
    {
      "question_text": "What is the chemical symbol for water?",
      "question_type": "one word",
      "options": {
        "answer": "H2O",
        "other_options": []
      }
    },
    {
      "question_text": "Which gas is responsible for the Earth's
greenhouse effect?",
      "question_type": "mcq",
      "options": {
        "answer": "Carbon dioxide",
        "other_options": ["Oxygen", "Nitrogen", "Methane"]
      }
    }
  ]
}
]
}

```

- **Response:**
 - **Success:**
 - Status Code: 201 Created
 - Body: A success message

Field	Type	Description	Example
message	String	A success message	Questions added successfully

Example:

```

{
  "message": "Questions added successfully"
}

```

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 409 Conflict, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

20. PUT /categories/questions/{question_id}

- **Description:** Update a question text
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:**
 - question_id: String (Path parameter, ID of the category to update)
- **Request Body:**

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>	<i>Required</i>
<i>question_text</i>	String	Updated question text	"What is the chemical formula of water?"	Yes

Example:

```
{  
  "question_text": "What is the chemical formula of water?"  
}
```

- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Question updated successfully

Example:

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

- **Error:**
 - Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

21. DELETE /categories/question/{question_id}

- **Description:** Delete a question
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:**
 - question_id: String (Path parameter, ID of the question to be deleted)
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: A success message

<i>Field</i>	<i>Type</i>	<i>Description</i>	<i>Example</i>
<i>message</i>	String	A success message	Question deleted successfully

Example:

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

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

22. GET /quiz?category_id={category_id}&question_type{question_type}&limit={limit}

- **Description:** Get random questions for quiz in a specified category if query parameter specified else across all categories
- **Headers:**
 - Authorization: **<Bearer token>** (required)
- **Parameters:**
 - category_id: String (Query parameter, ID of the category for the quiz)
 - question_type: String (Query parameter, mcq/ one word/ t/f types)
 - limit: Integer (Query parameter, number of questions for the quiz)
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: (Response containing question details for the specified category or random questions)

Field	Type	Description	Example
question_id	String	ID of the question	Q3cTnU
question_text	String	Question text	Capital of India is?
question_type	String	Question type	MCQ
options	Array[String]	List of options	["a", "b", "c", "d"]

Example:

```
[
  {
    "question_id": "Q3cTnU",
    "question_text": "Chemical formula for water is?",
    "question_type": "ONE WORD",
    "options": []
  },
  {
    "question_id": "Qz8Wno",
    "question_text": "Which gas causes greenhouse effect?",
    "question_type": "MCQ",
    "options": ["CO2", "Oxygen", "Nitrogen", "Methane"]
  }
]
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

23. POST /quiz/answers

- **Description:** Post player responses to the questions
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Request Body:**

Field	Type	Description	Example	Required
question_id	String	ID of the question	"QdeSe3"	Yes
user_answer	String	user's response	"Paris"	Yes

Example:

```
[
  {
    "question_id": "Q3cTnU",
    "user_answer": "H2O",
  },
  {
    "question_id": "Q4xZnP",
    "user_answer": "Carbon dioxide"
  }
]
```

- **Response:**
 - **Success:**
 - Status Code: 201 Created
 - Body: Player's score and responses

Field	Type	Description	Example
score	Integer	score achieved	90
responses	Array	Array responses	See example
question_id	String	ID of the question	"Q3cTnU"
question_text	String	Question text	"Water chemical formula?"
user_answer	String	User's answer	"H2O"
correct_answer	String	Correct answer	"H2O2"
is_correct	Boolean	Correctness of user's answer	false

Example:

```
{
  "score": 100,
  "responses": [
    {
      "question_id": "Q3cTnU",
      "question_text": "Chemical formula of water is?",
      "user_answer": "H2O",
      "correct_answer": "H2O2",
      "is_correct": false
    },
    {
      "question_id": "Q4xZnP",
      "question_text": "Chemical formula of Carbon Dioxide?",
      "user_answer": "CO2",
      "correct_answer": "CO2",
      "is_correct": true
    }
  ]
}
```

- **Error:**

- Status Code: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 500 Internal Server Error
- Body: Refer to Table 1.1 Status Codes

24. GET /leaderboard

- **Description:** Get leaderboard details
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:** None
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: (Response containing leaderboard details)

Field	Type	Description	Example
player_id	String	ID of the player	"P3qZnR"
username	String	Player's username	"john_wick"
score	Integer	Player's score	90
timestamp	String	Timestamp of submission	"2024-01-15 18:12:20"

Example:

```
[
  {
    "player_id": "P3qZnR",
    "username": "john_wick",
    "score ": "90",
    "timestamp ": "2024-01-15 18:12:20"
  },
  {
    "player_id": "P3qZnR",
    "username": "charlie_cox ",
    "score ": "80",
    "timestamp ": "2024-01-15 18:12:20"
  },
]
```

- **Error:**
 - Status Code: 401 Unauthorized, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

25. GET /scores/me

- **Description:** Get player's past scores
- **Headers:**
 - Authorization: <Bearer token> (required)
- **Parameters:**
 - player_id: String (Path parameter, ID of the player to retrieve the score)
- **Response:**
 - **Success:**
 - Status Code: 200 OK
 - Body: (Response containing the player's score details)

Field	Type	Description	Example
score_id	String	ID of the score	"S3qZnR"
score	Integer	Player's score	90
timestamp	String	Timestamp of submission	"2024-01-15 18:12:20"

Example:

```
[
  {
    "score_id": "S3cTnU",
    "score": "90",
    "timestamp": "2024-01-15 18:12:20"
  },
  {
    "score_id": "S3cTnU",
    "score": "80",
    "timestamp": "2024-01-14 17:12:20"
  }
]
```

- **Error:**
 - Status Code: 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error
 - Body: Refer to Table 1.1 Status Codes

Status Codes:

<i>Status Code</i>	<i>Status</i>	<i>Description</i>	<i>Example Message</i>
200	OK	Success of the request	<Requested data>
201	Created	A resource is created	Successfully Created
400	Bad Request	Incorrect data format of the request body	Invalid request format
401	Unauthorized	Invalid credentials or token expired	Please log in
403	Forbidden	Access is restricted to the resource as per the authenticated role	Access denied
404	Not Found	Resource does not exist	Data not found
409	Conflict	Resource already exists and an attempt is made to create it	Entity already exists
422	Unprocessable Content	Validation failure on the request body fields	Incorrect field type
500	Internal Server Error	Some error at the server	Something went wrong

Table 1.1 Status Codes