AI Quizzer API Documentation

Base URL

```
http://localhost:4000
```

Authentication

The API uses JWT (JSON Web Token) for authentication. Include the token in the `Authorization` header for protected endpoints.

Endpoints

1. Authentication

Login

POST `/api/auth/login`

Authenticate a user and receive a JWT token.

Request Body:

```json
{
  "username": "student1"
}
```

Response:

```json
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "username": "student1",
    "userId": "68d83c55d886bc34f816a034"
  }
}
```

---

2. Quiz Generation

Generate Quiz

POST `/api/quizzes/generate`

Generate a new quiz based on specified parameters.

Headers:

```
Authorization: Bearer <jwt_token>
```

Request Body:

```json
{
  "grade": 5,
  "subject": "Maths",
  "totalQuestions": 5,
  "maxScore": 10
}
```

Parameters:

- `grade` (integer): Grade level (1-12)
- `subject` (string): Subject name (e.g., "Maths", "Science", "English")
- `totalQuestions` (integer): Number of questions to generate
- `maxScore` (integer): Maximum possible score for the quiz

Response:

```json
```

```
{
  "quizId": "68d8462be3a76fb8cb8e51ce",
  "questions": [
    {
      "questionId": "68d8462be3a76fb8cb8e51d0",
      "question": "What is 5 + 3?",
      "options": ["6", "7", "8", "9"],
      "type": "multiple-choice"
    }
  ],
  "totalQuestions": 5,
  "maxScore": 10
}
```

---

3. Quiz Interaction

Get Hint

GET `/api/quizzes/{quizId}/hint?questionId={questionId}`

Get a hint for a specific question in a quiz.

Parameters:

- `quizId` (path): The quiz identifier
- `questionId` (query): The question identifier

Example:

```
GET /api/quizzes/68d8462be3a76fb8cb8e51ce/hint?questionId=68d8462be3a76fb8cb8e51d0
```

Response:

```json
{
  "hint": "Think about basic addition. What happens when you add 5 and 3 together?"
}
```

Submit Quiz

POST `/api/quizzes/submit`

Submit answers for a quiz.

Request Body:

```json
{
  "quizId": "68d8462be3a76fb8cb8e51ce",
  "responses": [
    {
      "questionId": "68d8462be3a76fb8cb8e51d0",
      "userResponse": "B"
    },
    {
      "questionId": "68d8462be3a76fb8cb8e51d1",
      "userResponse": "C"
    }
  ]
}
```

Response:

```json
{
  "score": 8,
  "maxScore": 10,
  "percentage": 80,
  "results": [
    {
      "questionId": "68d8462be3a76fb8cb8e51d0",
      "correct": true,
      "userResponse": "B",
      "correctAnswer": "B"
    },
    {
      "questionId": "68d8462be3a76fb8cb8e51d1",
      "correct": false,
      "userResponse": "C",
      "correctAnswer": "D"
    }
  ]
}
```

Retry Quiz

POST `/api/quizzes/{quizId}/retry`

Retry a quiz with new answers.

Request Body:

```json
{
  "responses": [
    {
      "questionId": "68d8462be3a76fb8cb8e51d0",
      "userResponse": "C"
    },
    {
      "questionId": "68d8462be3a76fb8cb8e51d1",
      "userResponse": "D"
    }
  ]
}
```

Response: Same format as submit quiz response.

---

4. Quiz History

Get Quiz History

GET `/api/quizzes/history?grade={grade}&subject={subject}`

Retrieve quiz history for a specific grade and subject.

Query Parameters:

- `grade` (integer): Grade level filter
- `subject` (string): Subject filter

Example:

```
GET /api/quizzes/history?grade=5&subject=Maths
```

Response:

```json
```

```
{
  "quizzes": [
    {
      "quizId": "68d8462be3a76fb8cb8e51ce",
      "grade": 5,
      "subject": "Maths",
      "score": 8,
      "maxScore": 10,
      "percentage": 80,
      "completedAt": "2024-01-15T10:30:00Z",
      "attempts": 1
    }
  ],
  "totalQuizzes": 1,
  "averageScore": 80
}
```

---

5. Leaderboard

Get Leaderboard

GET `/api/leaderboard?grade={grade}&subject={subject}`

Get the leaderboard for a specific grade and subject.

Query Parameters:

- `grade` (integer): Grade level filter
- `subject` (string): Subject filter

Example:

```
GET /api/leaderboard?grade=5&subject=Maths
```

Response:

```
json
```

```
{
  "leaderboard": [
    {
      "rank": 1,
      "username": "student1",
      "totalScore": 95,
      "averageScore": 95,
      "quizzesCompleted": 1
    },
    {
      "rank": 2,
      "username": "student2",
      "totalScore": 85,
      "averageScore": 85,
      "quizzesCompleted": 1
    }
  ]
}
```

Error Responses

All endpoints may return the following error responses:

400 Bad Request

```json
{
  "error": "Bad Request",
  "message": "Invalid request parameters",
  "details": "Grade must be between 1 and 12"
}
```

401 Unauthorized

```json
{
  "error": "Unauthorized",
  "message": "Invalid or missing authentication token"
}
```

404 Not Found

```json
{
  "error": "Not Found",
  "message": "Quiz not found"
}
```

500 Internal Server Error

```json
{
  "error": "Internal Server Error",
  "message": "An unexpected error occurred"
}
```

Usage Examples

Complete Flow Example

1. **Login:**

```bash
curl -X POST http://localhost:4000/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"username": "student1"}'
```

2. **Generate Quiz:**

```bash
curl -X POST http://localhost:4000/api/quizzes/generate \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -d '{
    "grade": 5,
    "subject": "Maths",
    "totalQuestions": 5,
    "maxScore": 10
  }'
```

3. **Get Hint:**

```bash
curl -X GET "http://localhost:4000/api/quizzes/QUIZ_ID/hint?questionId=QUESTION_ID"
```

4. **Submit Quiz:**

```bash
curl -X POST http://localhost:4000/api/quizzes/submit \
  -H "Content-Type: application/json" \
  -d '{
    "quizId": "QUIZ_ID",
    "responses": [
      {"questionId": "QUESTION_ID_1", "userResponse": "A"},
      {"questionId": "QUESTION_ID_2", "userResponse": "B"}
    ]
  }'
```

Rate Limits

- **Authentication:** 10 requests per minute per IP
- **Quiz Generation:** 5 requests per minute per user
- **Other endpoints:** 60 requests per minute per user

Notes

- All timestamps are in ISO 8601 format (UTC)
- Quiz IDs and Question IDs are MongoDB ObjectIDs
- Response options are typically labeled as "A", "B", "C", "D"
- JWT tokens expire after 7 days
- Maximum of 20 questions per quiz
- Supported subjects: Maths, Science, English, History, Geography