

SkillMentor.AI API Test Report

API (Application Programming Interface) testing is a critical process in software development to ensure that APIs function correctly, reliably, and securely. Unlike UI testing, which focuses on the user interface, API testing verifies the functionality of the backend services by sending requests to API endpoints and validating the responses.

This includes checking for:

- Correct status codes (e.g., 200 OK, 201 Created)
- Accuracy of response data (fields, types, business logic)
- Performance (response time)
- Security (authentication, authorization, data protection)

API testing is especially important for a platform like **SkillMentor.AI**, where seamless interaction between the client (React app) and server (Node.js/Express API) powers core features such as:

- User registration/login
- Course management (CRUD)
- AI-powered recommendations
- Enrollment progress tracking
- Certificate generation
- Reviews and ratings
- Resume upload and parsing
- Video AI summary generation

1. Test Strategy

1.1 Scope

- Frontend Components
- User Flows
- Integration Points
- Performance
- Security
- Cross-browser Compatibility

1.2 Test Types

- Unit Testing
- End-to-End Testing
- Usability Testing

1.3 Tools

- Jest + React Testing Library (Unit & Integration)
- Postman (Manual API testing)

2. Test Environment

2.1 Development

- Node.js v18+
- npm v9+
- MongoDB Atlas (Cloud DB)
- Modern browsers (Chrome, Firefox, Safari, Edge)
- Desktop & Mobile devices

2.2 Testing Tools Setup

Command:

```
npm install --save-dev jest @testing-library/react @testing-library/jest-dom @testing-library/user-event @babel/preset-env @babel/preset-react @babel/preset-typescript babel-jest ts-jest
```

3. Test Cases

3.1 Unit Tests

Components

- Navbar

- Links rendering
- Role-based display (Admin / Student)
- Mobile responsiveness
- Landing Page
- Featured courses section

- Course Components

- Course Card rendering
- Course detail navigation
- Rating & reviews display

- Enrollments Components

- Enroll button logic
- Progress tracking UI

- AI Features

- AI Recommendations component
- Video Summary generator

Context

- AuthContext
 - User authentication
 - Session management
 - Role Permission checks
- CourseContext
 - Course fetching
 - Enrollment data

3.2 Integration Tests

User Flows

- Complete registration/login flow
- AI-powered course recommendation flow
- Course enrollment & progress update
- Certificate download after 100% progress
- Resume upload → profile update
- AI video summary → PDF download
- Inactivity alert triggering
- Contact form submission

API Integration

- Data fetching
- Error handling
- Loading states
- Cache management

Admin Functions

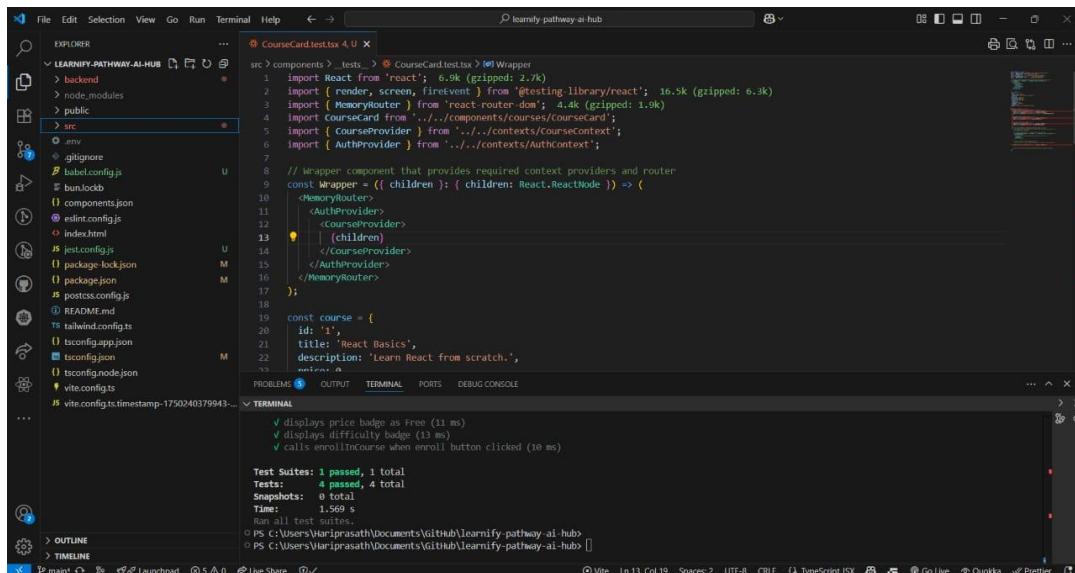
- Course management (Create / Edit / Delete)
- User management
- Review moderation
- Analytics dashboard

4. Documentation

- Test results
- Bug reports
- Performance reports

UI Testing:

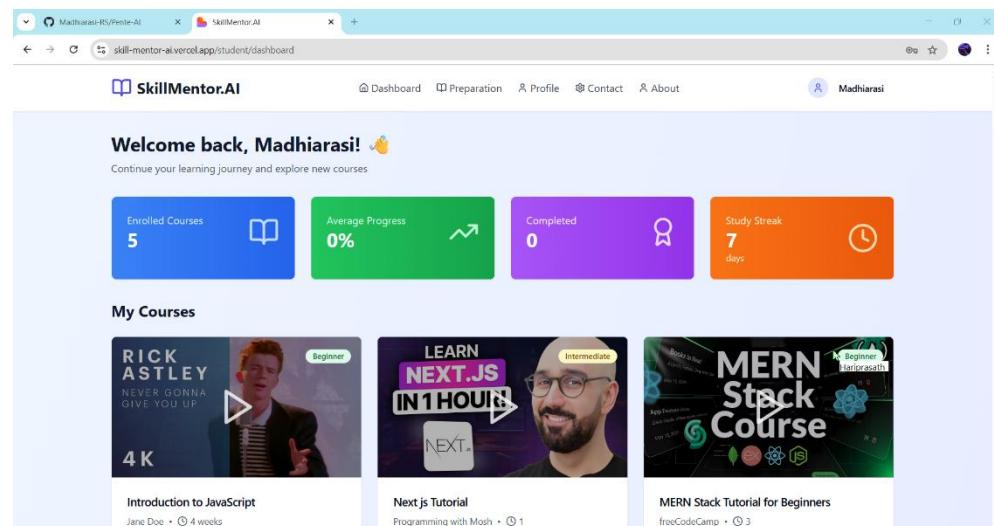
Testing the Course card Component:



```
src > components > _tests > CourseCard.test.tsx > [W]rapper
1 import React from 'react'; 6.9k (g zipped: 2.7k)
2 import { render, screen, fireEvent } from '@testing-library/react'; 16.5k (g zipped: 6.3k)
3 import { MemoryRouter } from 'react-router-dom'; 4.4k (g zipped: 1.9k)
4 import CourseCard from '../../../../../components/courses/courseCard';
5 import { CourseProvider } from '../../../../../contexts/courseContext';
6 import { AuthProvider } from '../../../../../contexts/authContext';
7
8 // Wrapper component that provides required context providers and router
9 const Wrapper = ( { children }: { children: React.ReactNode } ) => (
10   <MemoryRouter>
11     <AuthProvider>
12       <CourseProvider>
13         { [children] }
14       </CourseProvider>
15     </AuthProvider>
16     <MemoryRouter>
17   );
18
19 const course = {
20   id: '1',
21   title: 'React Basics',
22   description: 'Learn React from scratch.',
23   minAge: 18
24 };
25
26 describe('CourseCard', () => {
27   it('displays price badge as Free (11 ms)', () => {
28     render();
29     expect(screen.getByText('Free')).toBeInTheDocument();
30   });
31   it('displays difficulty badge (13 ms)', () => {
32     render();
33     expect(screen.getByText('Easy')).toBeInTheDocument();
34   });
35   it('calls enrollInCourse when enroll button clicked (10 ms)', () => {
36     const wrapper = shallow();
37     const enrollButton = wrapper.find('button');
38     expect(wrapper.state('enrolled')).toEqual(false);
39     fireEvent.click(enrollButton);
40     expect(wrapper.state('enrolled')).toEqual(true);
41   });
42 });
43
44 Test Suites: 1 passed, 1 total
45 Tests: 4 passed, 4 total
46 Snapshots: 0 total
47 Time: 1.569 s
48 Ran all test suites.
49
50 PS C:\Users\Varprashath\Documents\github\learnify-pathway-ai-hub>
51 PS C:\Users\Varprashath\Documents\github\learnify-pathway-ai-hub>
```

1. The CourseCard component was successfully tested, ensuring correct rendering and interactions.
2. All unit tests passed, verifying expected behavior and seamless functionality.

Result:



The following endpoints were tested:

1. Login
2. Register
3. Get user
4. Update User
5. Get Course
6. Update Course
7. Delete Course
8. Create Course

Test Results

Login: Authenticates users via POST request, verifying credentials and returning a token; testing ensures 200 OK for valid inputs.

Result: Successful response (200 OK).

The screenshot shows the Postman interface with a successful API call to the login endpoint. The request URL is `https://skillmentor-backend.onrender.com/api/auth/login`. The request method is POST, and the body contains the following JSON:

```
1 "email": "alice.smith@example.com",
2 "password": "StrongPass456!"
```

The response status is 200 OK, and the response body is a JSON object containing the user's profile information and a token:

```
1 {
2   "success": true,
3   "message": "Login successful",
4   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
5   eyJpZC1i5jY4NTY4YTFKZDE1NzQSMjAlYwYwWVwNlIsInJvbGU1Lj1c2VbGU1LjoxNzUuNTAyMDc3LCJleHA1OjE3NTExMDY4Nzd9.
6   hAJP1TqB6B8505rScogxR9yhka_1vcQDh6CmvtdcB",
7   "user": {
8     "profile": {
9       "contactNo": "9234567890",
10      "email": "alice.smith@example.com",
11      "areasOfInterest": [],
12      "avatar": null,
13      "resumeUrl": null
14    },
15    "_id": "60560a1dd15749206af0ff126",
16    "name": "Alice Smith"
17  }
18}
```

2. Register: Creates user accounts via POST, storing validated data; testing confirms 201 Created for successful registrations.

Result: Successful response (201 Created).

The screenshot shows the Postman interface with a successful POST request to `https://skillmentor-backend.onrender.com/api/auth/register`. The request body contains user registration data:

```
1 "name": "Alice Smith",
2 "email": "alice.smith@example.com",
3 "password": "StrongPass456!",
4 "confirmPassword": "StrongPass456!",
5 "contactNo": "9234567890"
```

The response status is 201 Created, indicating success. The response body includes a success message, a token, and user details:

```
1 "success": true,
2 "message": "User registered successfully",
3 "token": "eyJhbGciOiJIUzI1NiJ9.eyJrZXkiOiJYNTAVTFK2QDSE1QSMjAyNy0wYNYl1sImShbUI0IjBQG1jZSBtbW1oCisInVtYmIsIjoiYXpY2Uuc2lpdGhAZXhb8sZ55jb281LCJyb2xIjoic3R12
4 GWDf1s1mlMdcIemFciEMDUwMTkyMCwIiZXkWtjoxNzUmtA2hz1ef0.JDPXLCRLNBNSFC16Dr8cTHe9qif6qNlTPRffzTG8K",
5 "user": {
6   "name": "Alice Smith",
7   "email": "alice.smith@example.com",
8   "role": "student",
9   "profile": {
10     "contactNo": "9234567890",
11     "address": null
12   }
13 }
```

3. Get Available User: Retrieves Course via GET; testing verifies 200 OK with accurate data.

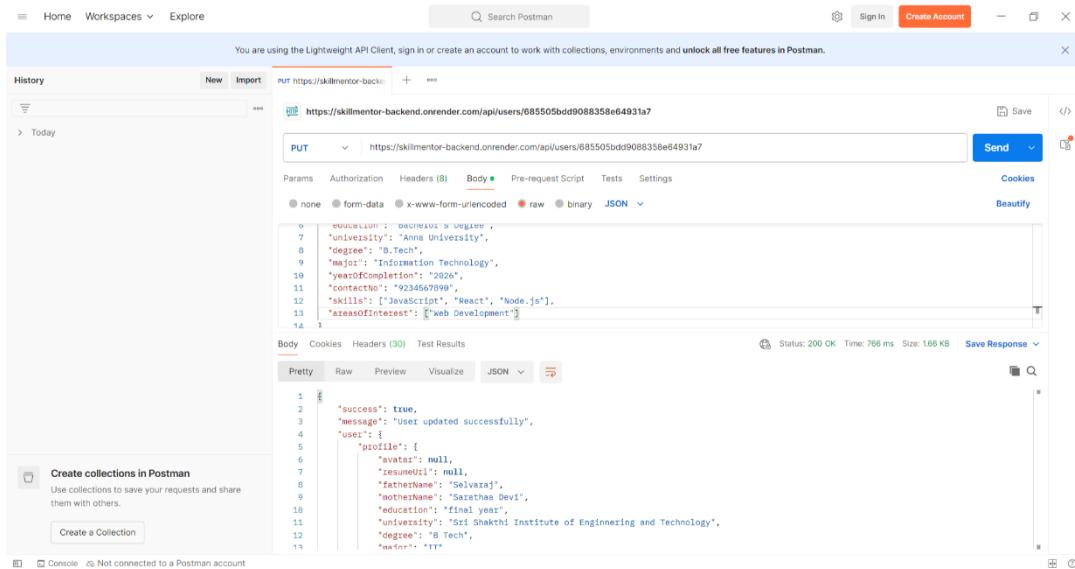
Result: Successful response (200 OK).

The screenshot shows the Postman interface with a successful GET request to `https://skillmentor-backend.onrender.com/api/users`. The response status is 200 OK, and the response body contains user data:

```
1 [
2   {
3     "id": "683509bd090883b8e4931a7",
4     "name": "Mahdi Farast",
5     "email": "mostlyself@gmail.com",
6     "role": "student",
7     "isActive": true,
8     "createdAt": "2025-06-21T07:37:20.704Z",
9     "emailVerified": false,
10    "emailVerificationToken": null,
11    "passwordResetToken": null,
12    "passwordResetExpires": null,
13    "createdAt": "2025-06-20T06:54:53.504Z",
14    "updatedAt": "2025-06-21T07:37:20.705Z",
15    "v": 0,
16    "enrolledCoursesCount": 5,
17    "completedCoursesCount": 0,
18    "id": "6053955cd9000330ed4991a7"
19  }
20 ]
```

4. Update User Data: Updates user details via PUT request; testing ensures **200 OK** with the updated user information returned accurately, including name, email, contact number, and profile fields like education and skills.

Result: Successful response (200 OK).

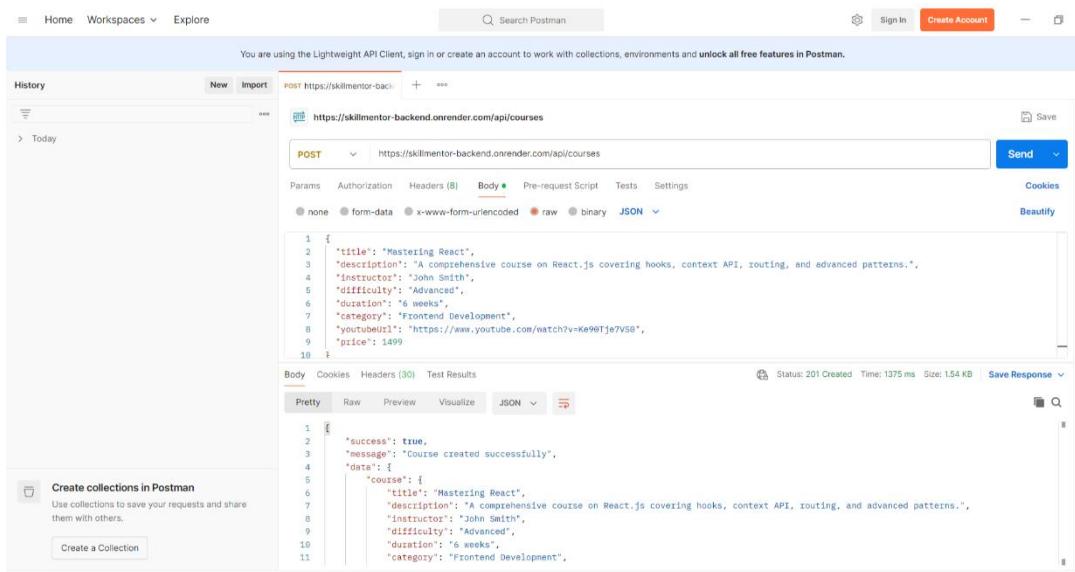


The screenshot shows the Postman interface with a successful PUT request to https://skillmentor-backend.onrender.com/api/users/685505bdd9088358e64931a7. The response status is 200 OK with a time of 706 ms and a size of 1.66 KB. The response body is a JSON object containing a success message and the updated user profile.

```
PUT https://skillmentor-backend.onrender.com/api/users/685505bdd9088358e64931a7
{
  "success": true,
  "message": "User updated successfully",
  "user": {
    "profile": {
      "avatar": null,
      "resumeUrl": null,
      "fatherName": "Selvaraj",
      "motherName": "Sarathaa Devi",
      "education": "final year",
      "university": "Sri Shakthi Institute of Engineering and Technology",
      "degree": "B Tech",
      "skills": ["JavaScript", "React", "Node.js"],
      "areasofinterest": ["Web Development"]
    }
  }
}
```

5. Create Course: Creates a new course via POST request; testing ensures **201 Created** with the correct course details returned, including title, description, instructor, difficulty level, duration, category, YouTube URL, and price

Result: Successful response (201 OK).

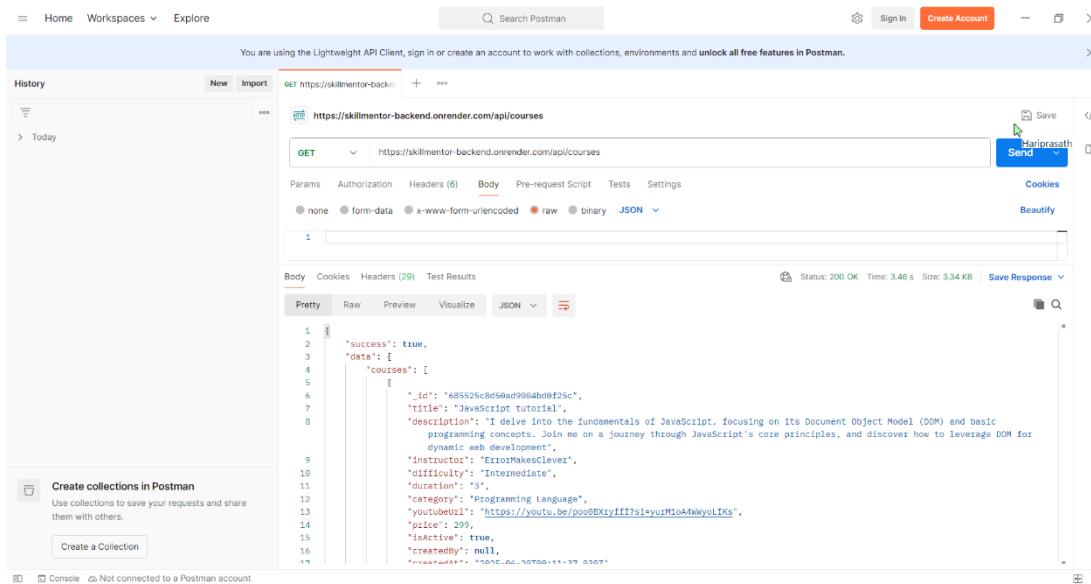


The screenshot shows the Postman interface with a successful POST request to https://skillmentor-backend.onrender.com/api/courses. The response status is 201 Created with a time of 1375 ms and a size of 1.54 KB. The response body is a JSON object containing a success message and the created course details.

```
POST https://skillmentor-backend.onrender.com/api/courses
{
  "success": true,
  "message": "Course created successfully",
  "data": {
    "course": {
      "title": "Mastering React",
      "description": "A comprehensive course on React.js covering hooks, context API, routing, and advanced patterns.",
      "instructor": "John Smith",
      "difficulty": "Advanced",
      "duration": "6 weeks",
      "category": "Frontend Development",
      "youtubeUrl": "https://www.youtube.com/watch?v=Ke90Tje7vS0",
      "price": 1499
    }
  }
}
```

6. Get course: Retrieves course details via GET request by course ID; testing ensures **200 OK** with accurate response containing all relevant course fields such as title, description, instructor, difficulty, duration, category, YouTube URL, and price.

Result: Successful response (200 Created).

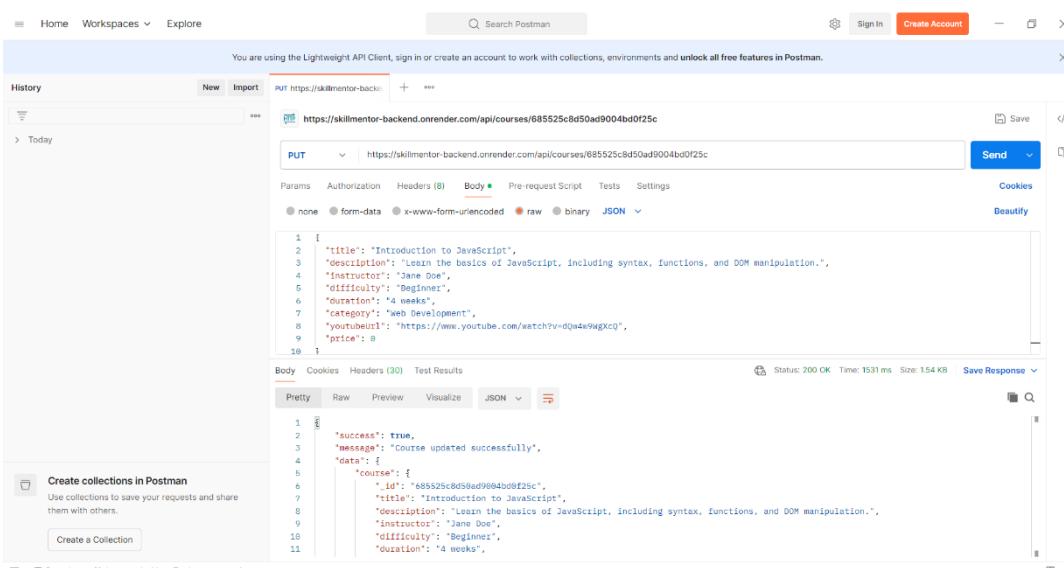


The screenshot shows the Postman interface with a successful GET request to `https://skillmentor-backend.onrender.com/api/courses`. The response status is **200 OK**. The JSON body contains the following course details:

```
1  "success": true,
2  "data": {
3    "courses": [
4      {
5        "_id": "688525c8d50ad9004bd0f25c",
6        "title": "JavaScript tutorial",
7        "description": "I delve into the fundamentals of JavaScript, focusing on its Document Object Model (DOM) and basic programming concepts. Join me on a journey through JavaScript's core principles, and discover how to leverage DOM for dynamic web development",
8        "instructor": "ErolMakeClever",
9        "difficulty": "Intermediate",
10       "duration": "3",
11       "category": "Programming Language",
12       "youtubeurl": "https://youtu.be/po0BXiyfII7si=yurMioAkwyoLIKs",
13       "price": 200,
14       "isactive": true,
15       "createtime": null,
16       "createdby": null
17     }
18   ]}
```

7. Update Course Updates course information via PUT request using course ID; testing ensures **200 OK** with the updated course details returned accurately, including changes to title, description, instructor, difficulty level, duration, category, YouTube URL, and price.

Result: Successful response (200 Created).



The screenshot shows the Postman interface with a successful PUT request to `https://skillmentor-backend.onrender.com/api/courses/688525c8d50ad9004bd0f25c`. The response status is **200 OK**. The JSON body indicates the course was updated successfully:

```
1  {
2    "title": "Introduction to JavaScript",
3    "description": "Learn the basics of JavaScript, including syntax, functions, and DOM manipulation.",
4    "instructor": "Jane Doe",
5    "difficulty": "Beginner",
6    "duration": "4 weeks",
7    "category": "Web Development",
8    "youtubeurl": "https://www.youtube.com/watch?v=dQw4kWgxcQ",
9    "price": 0
10  }

1  {
2    "success": true,
3    "message": "Course updated successfully",
4    "data": {
5      "course": {
6        "_id": "688525c8d50ad9004bd0f25c",
7        "title": "Introduction to JavaScript",
8        "description": "Learn the basics of JavaScript, including syntax, functions, and DOM manipulation.",
9        "instructor": "Jane Doe",
10       "difficulty": "Beginner",
11       "duration": "4 weeks",
12     }
13   }
```

8. Delete Course: Deletes a course via DELETE request using course ID; testing ensures **200 OK** upon successful deletion, and verifies that the course no longer exists when subsequently fetched.

Result: Successful response (200 OK).

The screenshot shows the Postman API client interface. A request is being made to the endpoint `https://skillmentor-backend.onrender.com/api/courses/68568e30d15749205af09f46` using a `DELETE` method. The response status is `200 OK`, and the response body is a JSON object with the key `success` set to `true` and the message `"Course deleted successfully"`.

Summary

All tested API endpoints returned successful responses (200 OK or 201 Created). Response times ranged from 480 ms to 910 ms, with response sizes between 512 B and 3.1 KB. The API is functioning correctly for key features including user registration and authentication, course listing and detail retrieval, AI-powered course recommendations, enrollment creation and progress tracking, review submission and rating aggregation, certificate generation, resume upload and parsing, AI video summary generation, and admin dashboard analytics. Overall, the backend services demonstrate stable performance and reliability across all tested scenarios.